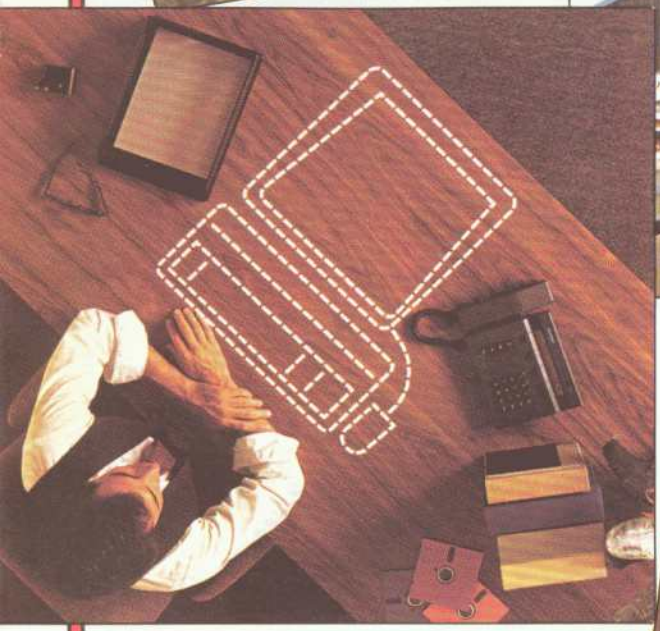
μ mikro számítógép magazin

Ára: 36 Ft

1987/2

Az új színhely

SICOB



μ mikro számítógép magazin

5. ÉVFOLYAM
1987/2. SZÁM

A NEUMANN JÁNOS SZÁMÍTÓGÉPTUDOMÁNYI TÁRSASÁG ÉS A KISZ KÖZPONTI BIZOTTSÁG LAPJA

A kiadvány a Tudományszervezési és Informatikai Intézettel együttműködve készül

A szerkesztőbizottság vezetője:
Kovács Győző

E számunkat szerkesztették:

Bakos Tamás
(programozástechnika)

Broczkó Péter
(hírek)

Kovács Győző
(levelezés)

Lindner László
(sakkprogramozás)

Petróczy Judit
(könyvek)

Simonyi Endre
(klub)

Varga András
(iskola-számítógép)

**Címképünk:
Perényi József munkája**

**Hátsó borítónk képe
Kozák Gábor munkája.
Olvasóink hasonló jellegű képeit, rajzait, grafikáit szívesen fogadja a szerkesztőség**

Felelős szerkesztő:
Könyves Tóth Pál

Szerkesztőség:
1027 Budapest, Fő u. 68.
Telefon: 154-250

Levélcím
1371 Budapest
Pf. 433.

Kiadja az Ifjúsági Lap- és Könyvkiadó Vállalat

Felelős kiadó:
dr. Petrus György
igazgató

Kiadóhivatal:
1065 Budapest, Révay u. 16.
Telefon: 116-660

Terjeszti a Magyar Posta
Előfizethető a hírlapkézbesítő hivataloknál
és a Posta Hírlapelőfizetési és Lapellátási Irodáján
(1900 Budapest V., József nádor tér 1.)
vagy átutalással a 215—96 162 pénzforgalmi jelzőszámra.

Megjelenik havonta
Egy szám ára 30,— Ft
Előfizetési díj:
egy évre 360,— Ft
fél évre 180,— Ft
Külföldön terjeszti a Kultúra,
1389 Budapest, pf. 149.
és a Magyar Média
1932 Budapest, pf. 279.
86—253

Szikra Lapnyomda
Budapest (86-4731)
Felelős vezető:
Csöndes Zoltán vezérigazgató

INDEX: 25 629
ISSN 0236-6088

Tartalom

Alkalmazás '86 2
A programozás Trabantja 14
Vigyázat! Tolvaj! 15
Rajzológépek 16
SICOB — volt és lesz! 18
Adok-veszek-cserélek 19
μINFORM 27
Interaktív kompakt lemezek 39

ISKOLA — SZÁMÍTÓGÉP

Meghalt a király, ki lesz az új király? 3
Ismerős gondok az Egyesült Államokban 4
Másolóprogram 8
PRIMO-primátus 8
Gépi kódban 9
Tv-foci 10

PROGRAMOZÁSTECHNIKA

Z80 programozási gyakorlatok 7. 12
BASIC és gépi kód 13

μKLUB

QLub 20
Adom a magyarázatot! 20
A Spectrum operációs rendszere ZX81-en 21

μPROGRAMOK

A Beta BASIC és alkalmazása III. 30
Mandelbrot-program 32
VC20 programok beolvasása 34
Floppy monitor 35
Disassembler program 36
HELP bővítő 37

SAKKPROGRAMOZÁS

Bitek és figurák 38

JÁTÉKPROGRAMOK

Kétórás játékok — miért ne? 40
Labirintus 41
Kapcsoltam? 42

AZ OLVASÓ ÍRJA 43

KÖNYVEK 45

HÍREK-ÉRDEKESSÉGEK 47

Alkalmazás '86

A Neumann János Számítógéptudományi Társaság III. Országos Kongresszusa Szolnok, 1986. november 17—20.

„Kiváló emberekhez valóban nem méltó, hogy rabszolga módra órákat vesztegessenek el olyan számítások elvégzésével, amelyeket bárkire nyugodtan rá lehetne bízni, ha gépet használna."
(Gottfried Wilhelm Leibniz — 1646-1716)

A kedves olvasó bizonyára furcsállja, hogy egy múlt évi eseményről négy hónap után írok, de a nyomdai átfutási idő mindenre magyarázatot ad. Amikor megpróbálom az országos kongresszus tapasztalatait a számítástechnika iránt érdeklődő olvasók részére összefoglalni, még egy hete sincs, hogy a kongresszust befejeztük, így meglehetősen frissek az élményeim, és talán februárra sem avulnak el.

Szolnok fogadta az Alkalmazás '86 több, mint négyszáz résztvevőjét, szívélyesen és barátsággal; a budapesti és helyi rendezők mindent megtettek az összejövetel sikere érdekében. Három teljes nap alatt 96 előadás hangzott el négy szekcióban. Így — mondta összefoglalójában Gergely Csaba, aki a program- és a szervezőbizottságot is képviselte — még a legszorgalmasabb résztvevő is legfeljebb az előadások egynegyedét hallgathatta végig. Én azt hiszem, hogy a rendezők jól tették, hogy szinte valamennyi előadót a pódiumra engedtek, nem csak a „sztár" előadókat, de a sokszor motyogó és nem kevésszer az előadást felolvasó kezdőket is. Amikor jó néhány évvel ezelőtt a Neumann-kongresszusok megrendezését javasoltam, az egyik célom éppen az volt, hogy a kezdők is lehetőséget kapjanak az előadói készség és gyakorlat megszerzésére. Erre itt a lehetőség megvolt, legfeljebb az idő volt kevés (pl. az oktatási szekcióban). Szerencsére jó volt a hallgatóság, nagyon sokszor vitatkozott, vagy szeretett volna vitatkozni az előadókkal, ha nem vágta volna el a szigorú elnöki szó a vita fonalát.

A kongresszus másik fontos célja — véleményem szerint — a találkozás volt. Nagyon sokan eljöttek, akiket az ember — a munkája során — csak nagyon ritkán, vagy sohasem lát. Sajnos sokan nem jöttek el, mert vagy az anyagiak hiányoztak (voltaképpen drága négy napot töltöttünk Szolnokon, hiszen 3500 Ft volt a részvételi díj, 1—2000 Ft a szállás, 4—500 Ft volt az étkezés, de — sajnos — ennyiért lehet ma egy ilyen összejövetelt megrendezni), vagy pedig a kongresszus témája nem találkozott jó néhány Neumann-tag érdeklődésével. A kongresszus záróülésén azt javasoltam, hogy ezután valamennyi kongresszust nevezzük „Alkalmazás"-nak, de a felhívásban hívjuk meg előadóként a hardver és szoftver konstruktőröket is, ne csak az alkalmazókat, hiszen az ő munkájuk is része, sőt fontos része az alkalmazói rendszereknek. Az sem baj, ha nem négy, hanem nyolc szekció fut párhuzamosan, az a lényeg, hogy a tagságnak minél nagyobb része ott legyen ezen a háromévenkénti összejövetelen. Ebből a szempontból a megelőző kongresszus témaválasztása még szerencsétlenebb volt (Adatbázisokon alapuló információs rendszerek), mert a szakembereknek egy igen szűk rétegéhez szólt, így nagyon sokan voltak távol.

A szekcióelnökök referátumából feljegyeztem néhány gondolatot, amelyeket — némi kommentárral — most közreadok.

— Elég sok volt az igazán jó és tartalmas előadás, de nem volt kevés az olyan sem, ami nagyon nagy munkával előállított szerénynek eredményről adott számot. A legfeltűnőbb talán az volt — az előadásokból világosan látszott —, hogy nálunk csak a saját fejlesztésű rendszerekről illik (sikk!) beszámolni, olyasmiről nem, amikor pl. valaki egy kész rendszert (dBASE, . . . calc stb.) alkalmaz egy feladat megoldására. Nem hiszem, hogy kiszámolta már valaki, hogy mennyi időt, pénzt és energiát használunk el feleslegesen, mert vagy nem ismerjük a kész és konfekcionált rendszereket, vagy pedig egyszerűen nem akarjuk használni ezeket, mert az nem elégíti ki teljes mértékben a megrendelők vagy éppen a rendszerszervező előzetes elképzeléseit. Dömölki Bálint mondta zárszavában, idézve egy, az IFIP kongresszuson elhangzott előadásból, hogy a jövőben, de már ma is az alkalmazás technológiája azt jelenti, hogy az alkalmazó kiválasztja az elképzeléseihez legközelebb álló rendszert, esetleg több ismert rendszerből próbál meg egy komplexet előállítani, és a saját szervezetét próbálja meg ehhez az alkalmazói rendszerhez passzítani.

Már a kongresszuson is suttogták, hogy ez a „gombhoz a kabát" technológia — lehet, de hogy a költségeket csak így lehet alacsony szinten tartani, abban teljesen biztos vagyok.

— Néhány előadás után úgy éreztem, mintha a költségek alacsony szinten tartása egyáltalán nem szerepelt volna az előadók által készített műszaki feladattervben. Kerül amibe kerül, legfeljebb magasabb lesz az üzem, a vállalat, az intézet költsége, a lényeg, hogy a rendszer menjen. Nemegyszer került az előadó meleg helyzetbe, amikor a kérdezők a megoldás költségeit feszegették.

— Viszonylag sok előadó mondta, hogy az általa bemutatott és esetenként nagyon kritikusan fogadott alkalmazói rendszer bevezetése után azt nagyon gyorsan befogadták és megszerették, különösen az interaktív programokat. Néhány hónap múlva el sem tudták képzelni a munkát számítógép nélkül. Én azt hiszem, és ebben a hitemben a kongresszus megerősített, hogy a hazai alkalmazók most jutottak el annak a felismerésére, hogy vannak olyan problémák, amelyeknek a megoldására eddig azért sem kerülhetett sor, mert nem volt eszközük hozzá. Egyre több olyan példát láttunk, amikor nem eddig kézi eszközökkel végzett munkát akartak gépesíteni, hanem teljesen új és manuálisan kezelhetetlen alkalmazói területeken próbáltak szerencsét.

— Számomra nagy és igen kellemes meglepetés volt, hogy az oktatási szekció iránt mennyien — és amint utólag kiderült, voltaképpen nem csak pedagógusok — érdeklődtek. A programbizottság egy megbocsátható tévedése volt a szekciót a kisebbik terembe tenni, ráadásul az előtte levő szekció „túlcsordult", így Szelezsán János szekcióelnök kénytelen volt az egyes előadások idejét 20 percre redukálni, ráadásul a vitát sem engedélyezte. (Bosszúságomban esti találkozóra hívtam a hallgatókat, akik — örömömre — el is jöttek, szinte ismét megtelt a terem, és így nagyon kellemes körülmények mellett, számomra igen tanulságos beszélgetésen hallgathattam meg a nyílt egyetem létrehozásával kapcsolatos bátorító véleményeket).

— Föltétlenül baj volt, hogy a „maszek kerekasztalomhoz" hasonló esti vagy délutáni találkozókat, beszélgetéseket nem szerveztünk. Pedig biztosan jól megtárgyalhattuk volna a már említett problémákat, de azt a sok, a szakmát izgató kérdést is, amelyek így a résztvevőkben „bennszorultak".

— A nyitó előadásokat Vámos Tibor, Pál László, Reiniger Péter és dr. Udovecz Gábor tartották, a műszaki fejlesztés, az ipar és a mezőgazdaság számítástechnikai, elektronikai terveiről és gondjairól beszéltek. Sajnos, ezeket az előadásokat sem követte vita; nagyon sokan vélték úgy, hogy talán a főhatóságoknak és a résztvevőknek is haszosabb lett volna egy ilyen beszélgetés, mint a kétségkívül kitűnő táncegyüttes műsora.

A Neumann-kongresszusok emlékezetes pillanata a Neumann-díjak és a Kalmár emlékérmek kiosztása. Az idén — ez a személyes véleményem — igen jól sikerült a díjak odaítélése, az erre érdemesek kapták a kitüntetést. Óriási taps köszöntötte a társaság tíz évig aktív, ma tiszteletbeli elnökét, Vámos Tibort, amikor a díjat átvette. Hasonló ünneplésben részesült Könyves Tóth Pál, aki a magyar számítástechnikai szaksajtóban végzett munkájáért és Jávor András, aki az első kórházi számítógépes rendszer kidolgozásáért vehette át a Neumann-díjat. A Kalmár emlékérmet főleg szakmai eredményekért ítéli meg a társaság országos elnöksége. Az idén Hanák Péter az egyetemi oktatásban, Lőcs Gyula a programozásban elért eredményéért és a két Lukács testvér — megosztott díjat — az első hazai házi számítógép kifejlesztéséért részesült az egész tagság elismerését kifejező kitüntetésekben.

A III. Országos Kongresszus befejeződött. Inkább a problémákról írtam, mert csak ebből tanulhatunk és rendezhetünk olyan kongresszusokat, amelyek egyre jobban összekötnek bennünket. Ezt kívánja valamennyiüknek:

KOVÁCS GYŐZŐ

Meghalt a király, ki lesz az új király?!

Az iskolákban közismert HT—1080Z hazai gyártmányú iskolaszámítógép elérkezett életpályája végéhez. Az utolsó darabok ez év elején hagyták el a gyártó műhelyeit, s a közelmúltban a maradék ötszázat is kiárusították, leszállított áron. Eddig 2364 darab került a különböző oktatási intézményekbe, s közülük 1861 kifejezetten az iskolákba.

Korai lenne mégis a búcsúztató: e gépekre ugyanis elég sok program készült az idők folyamán, s várhatóan egyes szervizek és gmk-k még évekig vállalni fogják a javítást. A gép egyik előnye, hogy egyszerű felépítésű, különleges alkatrészek nélkül épült. Ma már túlhaladott, de segítségével indult meg az iskolaszámítógépes program, vele szereztük az első tapasztalatokat az oktatásban, így érdemei feledhetetlenek.

A HT—1080Z a TRS—80 típusú nyugati mikrogép klónjába tartozik. A Microsoft BASIC level-two verzióit használja, így BASIC szinten — a hangrész kivételével — kompatibilis volt külhoni elődjével. Ugyanakkor a gyártás folyamán is megváltozott, így ennek a BASIC-nek, de főleg az assemblernek különböző dialektusai alakultak ki. Az első, 1983-as kiadáson a magnó felvételi szintjét még kézzel kellett állítani, s nem volt ékezetes betű sem. A következő évi sorozatnál már megjelentek egyes ékezetes betűk, sőt a szintszabályozást is automatizálták. Végül, 1985-ben megjelent a 64 kbájtos változat — még több magyar karakterrel. De ez is képtelen volt a színes, finomgrafikus képmegjelenítésre, ami a korszerűbb személyi számítógépeknél, főleg, ha az oktatásban alkalmazták őket, elengedhetetlen.

De mi legyen ezután? Ennek a fogas kérdésnek az eldöntésére az iskolaszámítógép-program koordinátora, a Tudományszervezési és Informatikai Intézet pályázatot írt ki. Ezen a megmérettetésen négy gép bizonyult alkalmasnak arra, hogy a nebulók oktatását szolgálja. Mint azonban majd látni fogjuk, valószínűleg csak egy hazai és egy külföldi típus kerül be ténylegesen az oktatási intézmények falai közé.

Elöljáróban azonban érdemes egy pár szót szólni az árakról. A cikkben szereplő árak nem azonosak azokkal, amelyekért az üzletben bárki (nem oktatási intézmény) megvásárolhatja ezeket a berendezéseket. Külföldön már régen bevált az a gyakorlat, hogy az oktatási intézmények igen kedvezményes áron, forgalmiadó- és vámmentesen kapják a berendezéseket. Hasonló feltételeket kell itthon is biztosítani. Az árak értékeléséhez azonban azt is tudni kell, hogy amennyiben ezeket a készülékeket az iskolák a TII-n keresztül szerzik be, további egyharmaddal olcsóbban juthatnak hozzá, mert a különbözetet az iskolaszámítógép-program gazdái költségvetési forrásokból fedezik.

Az egyik jövendő új iskolagépnek a HT3080C látszott, de kétséges, hogy sorozatgyártása egyáltalán megindul-e. Maga a gép nem kompatibilis az őt megelőző HT iskolagép-sorozat egyik tagjával sem. Lényegében a ZX-Spectrum programjai futtathatók rajta, így természetesen finomgrafikus lehetőséggel és színes megjelenítési móddal is rendelkezik. A BME kollégiumának számítástechnikai köre fejlesztette ki. Várható kedvezményes ára 27 ezer forint lenne; sokkal több, mint a trón más várományosaié. Csak igen nagy sorozatban lehetne olcsóbb, mert akkor érdemes lenne itthon kifejleszteni a berendezésorientált áramköröket és gyártásukat.

A másik jelölt a Primo utódgépe lett volna. Maga az eredeti alapgép — a billentyűs, illetve a fóliatasztatúrás — gyártása megszűnt, a maradék készleteket most árusítják ki árengedménnyel. A pályázaton szerepelt utód a PRO-Primo, amelynek van színes grafikája, és egy hazai BASIC „dialektust" alkalmaz. A megajánlott kedvezményes ár szintén magas: a 32 kbájtos változat 13 500, a 64 kbájtos pedig 17 ezer forintba került volna. Eddig csak 850 darabra van megrendelés, de a gyártó — a COSY — véleménye szerint csak minimálisan 10—15 ezres sorozatnál tarthatják ezt az árat, így nem érdemes belefogni a sorozatgyártásba.

A hazai utód, az egyik általánosan elterjedt iskolagép szinte biztosan a Videoton TV-Computere lesz. Az idén már 1000 darabot szállítottak le belőle, a 32 kbájtost 11 ezer, a 64 kbájtost 13 ezer forintos kedvezményes áron. A jövő évtől csak a 64 kbájtos gépet fogják szállítani, 12 ezer forintos áron, és ebből 5000 darab kerülhet az iskolákhoz. Egyelőre még nincs hozzá szoftver, mert a Videoton csak a hardver gyártásával kíván foglalkozni. Így még mintegy fél évet kell várni az első oktatási célú programcsomagok megjelenéséig. Az ellátást az OPI programirodája szervezi. A koordinátori munkát és a sokszorosítást ez esetben is a TII intézi. A VT TV-Computer előnye, hogy teljes rendszert lehet összeállítani belőle, az oktatási intézmények költségvetésébe beférő árakon. Jövőre megjelenik a floppy és sornyomtató — ez utóbbi Centronics interfésszel. A BRG által gyártott adatmagnó már megvan, kedvezményes ára 2900 Ft. Sikeres konstrukció, olyannyira, hogy Bécsben is lehet látni a számítástechnikai áruházak némelyikében, igaz, nem ezen az áron . . . (drágább).

A közkedvelt Commodore sem maradhatott ki az iskolagépek közül. A C64-es az ára miatt nem versenyképes, de a C16-tal kompatibilis Commodore Plus/4 magnóval együtt 7900 forintos kedvezményes áron került be, amelyre a TII-től beszerzett gépekre további egyharmad kedvezményt kapnak az iskolák. Kapható lesz a közeljövőben a Commodore 1551-es floppyval is — adatmagnó helyett —, 18 600 forintos kedvezményes oktatási áron. A Commodore gépekre igen sok szoftver készült az iskolákban. A Novotrade Octasoft és Deltasoft irodája szintén forgalmaz programokat ehhez a típushoz, közöttük kifejezetten oktatóprogramokat is. A TII és az OPI jóvoltából rövidesen várható a tananyaghoz kapcsolódó programok megjelenése.

A TII és az OPI megrendelésére készült programokat az Akadémiai Könyvkiadó Városház utcai Magiszter Könyvesboltjában, mintegy 200 forintos egységáron, programkazettán bárki megvásárolhatja. Megjegyzendő, hogy ezek a programok nem másolásvédettek. A Novotrade által forgalmazottak sem, de turbósítva vannak, így igen nehezen másolhatók.

A TII-n keresztül eddig 3400 darab C16-os számítógép került forgalomba; iskolába ebből 3004 darab. A C Plus/4-ből eddig 5300 jutott az iskolákba, és megrendelés van további 1920 floppys gépre. Mind a C Plus/4-ből, mind a C16-ból az országban beszerzési forrástól függően több verzió van forgalomban. Ezek annyiban különböznek, hogy a TII által forgalmazott gépekben a ROM magyar karakterkészletre van kicserélve, a külföldről behozott gépek pedig vagy angol, vagy/és német karakterkészlettel dolgoznak. Ez galibát okozhat néhány külföldi eredetű játékprogramnál, amely a kisbetűt is használja: egyes félgrafikus karakterek helyett ugyanis a magyar ábécé betűi jelennek meg. A magyar oktatóprogramoknak — mivel ezek természetesen magyar ékezetes betűket is alkalmaznak — a direkt külföldről behozott gépeken való futtatása okozhat gondot. Áthidalja ezt a Novotrade-nál megvásárolható, a gépbe betölthető magyar karakterkészletet generáló szoftver, bár hallani arról is, hogy egyesek a problémát hardverrel — egy másik ROM beépítésével — is megoldották. A C Plus/4 64 kbájtos, tehát eleget tesz a korszerű iskolagép követelményeinek, s azt már mondani sem kell, hogy alkalmas finomgrafikus megjelenítésre. További szempont, hogy a közben már régebben forgalomban lévő C16-osok is bővíthetők 64 kbájtra. Vagy külső bővítő hozzácsatolásával, ami a nagyobb áramfelvétel miatt tápegységmódosítást tesz szükségessé, vagy egyes gmk-k ugyanezt, hardver úton is egyszerűen és viszonylag olcsón elvégzik.

Hogy ki lesz tehát az új király az iskolákban? Az idő ad erre végleges választ; még sok minden történhet. Annak idején sokan nem hittek a HT sikerében. A jelenlegi helyzet kettős királyságra utal: néhány évig valószínűleg a VT TV-Computer és a C Plus/4 lesz az uralkodó az iskolák és egyéb oktatási intézmények kabinetjeiben.

KIS JÁNOS

Magasabb nívón

Ismerős gondok az Egyesült Államokban

Simonyi Endre (rovatszerkesztőnk) egyesült államokbeli útja során találkozott többek között William S. Wagner úrral, az IBM oktatási osztályának tanácsadójával — korábban a Santa Clara megye számítástechnikai vezető szakfelügyelője, a Számítógép-használó Oktatók (CUE) nevű szervezet elnökségi tagja —, aki közlés céljára rendelkezésére bocsátotta egy konferencián tett felszólalásának anyagát. Mivel az iskolaszámítógép-programunkban részt vevő tanároknak, a program irányítóinak érdemes megismerkedniük az előttünk járó országok tapasztalataival, eredményeivel, közreadjuk Wagner úr előadásának érdeklődésükre számot tartó részleteit. Örülnénk, ha olvasóink reflexiókat fűznének a közöltekhez.

Visszatekintés

Először is perspektívába helyezném a jelent a számítógépes korszak, a számítógépes oktatás kezdeteire való visszatekintéssel. Létezett 1952-ben egy csodálatos számítógép (de hát melyik számítógép nem csodálatos a maga idejében?), az IBM 704. Ez a gép nagy teljesítményű volt, gyors volt, és lehetőségeivel elbűvölte az embereket. Nem számított, hogy bérleti díja havi 40 000 dollár volt, hogy öt mérnök állandó jelenlétét igényelte, és hogy tizpercenként leállt. Nem számított, mert a gép addig nem tapasztalt műveleti sebességgel és teljesítménnyel dolgozott.

1966-ban egy kevéske FORTRAN-t oktattam — egy hétig az osztályomban, továbbá egy este az egyetem számítóközpontjában. A produktumunk egy bináris program volt, amely kétjegyű számokat adott össze, és kiírt egy nevet. Nem érdekes, hogy csak ennyi volt, a lényeg az, hogy programoztunk egy számítógépet.

1977-ben vásároltam négy számítógépet az iskolámnak. Darabjáért 4000 dollárt fizettem, de higgyék el, az iskola vezetősége mégis meg volt elégedve a vásárlással: 25 százalékkal olcsóbban vettük ugyanis, mint egy akkor korszerűnek számító, négyterminálos mikroszámítógépes rendszer ára. Ugyanakkor rendelkezésünkre állt négy gép, mindegyik egy-egy monochrom monitorral, 48 k-ra bővíthető 32 k RAM-mal, 70 k lemezkapacitással és BASIC programozási nyelvvel. Mondhatná bárki most, és igaza lenne, hogy ezek a számítógépek nemigen hasonlíthatók az Apple vagy JX gépekhez; mégis, én akkor nagyon boldog voltam, mert azok a gépek ott az osztályomban lényegesen nagyobb teljesítményűek voltak, mint a 25 évvel korábbi legjobb gépek.

Tradicionális, veterán matematikatanárként el voltam bűvölve a jó szerencsémtől: lehetőségem nyílt arra, hogy az addigiaktól eltérő módon oktassak! De nyilván Önök közül is sokaknak volt már ez a tapasztalata: a diákok egy része nem mutatott érdeklődést, és otthagyott a pácban. Elmarasztaltak emiatt, érdemtelenül. A többi diákban viszont annál nagyobb volt a buzgalom. A BASIC tanítása olyan helyzetet teremtett, amelyben a diákok vagy azok kis csoportjai egyénileg haladtak, próbáltak megoldásra jutni, és módszereiket illetőleg azonnali visszajelzést kaptak a számítógéptől.

1978-ban alkalmam volt előadást tartani a Második Nyugati-parti Számítógép Vásáron, San Franciscóban. Reméltem, hogy lesznek ott olyan pedagógusok, akikkel érdemes lesz találkoznom. Meg kellett tudnom, hogy vannak-e rajtam kívül, akik bosszús és örömteli órákat töltenek a gépek mellett, új oktatási módszereket fejlesztve. Meglepetésemre több mint kétszázan hallgatták meg egy ismeretlen főiskolai tanár első előadását... Amikor az előadásom során újabb találkozót javasoltam, kiderült, hogy nagyon is sokan vannak a hallgatók között számítógép-kedvelők. Ez végül is a Számítógép-használó Oktatók (CUE) szervezetének megalakításához vezetett. 1978-ra ugyanis a következő felismerések fogalmazódtak meg:

— nincs követhető tananyag;

— nincsenek szervezetek, konferenciák, újságok;

— nincs tanárképzés; a legtöbb egyetem rengeteg időt fecsérel a számítógépes oktatás mikéntjének feltárására; nincs számítógépes oktatásról szóló diploma és nincs is hol megszerezni;

— szoftver alig létezik, és a főllelhetők legtöbbjét tanárok vagy más amatőrök írták; nincs szakirányú kutatás;

— a nagy kiadók és hasonló intézmények mellőzik ezt a tárgykört;

— az illetékesek nem foglalnak állást, hogy támogatják-e az oktatásnak ezt az irányát vagy sem;

— a hosszú távú tervezésre csak elvétve van példa: kevés iskolának vagy iskolakörzetnek van elképzelése, hogy milyen irányba tartson a számítógépes oktatásban.

Minthogy valamennyi felvetett probléma megoldásra várt, gazdagon kínálkozott alkalom a világ újraformálására. És bár valamennyi területen értünk el eredményt, a számítógépes oktatás olyan fiatal terület, hogy itt még mindig nagyon sok a tennivaló; nekünk, tanároknak még mindig szép számban lesz módunk ebben jelentős szerepet vállalni.

A 70-es években mindenesetre úgy láttuk, hogy korunkat meghaladva előbbre járunk, teljesen magunkra hagyatva. Ez végső soron természetes is — gondoljanak csak a tinédzserek viselkedésére. És ahogy ők általában, mi is messze jártunk jövőbeli elképzeléseinkkel a valóságtól. Íme néhány példa.

- Azt vártuk, hogy a számítógépgyártó cégek szaporodni fognak — bár az IBM belépésére akkor még nem számítottunk. Jelenleg azonban jobbára csak négy géptípus: Apple II, TRS—80 Co-Co, Commodore 64 és IBM PC jellemző az egyesült államokbeli iskolák gépparkjára, azaz kevesebb, mint 1979-ben.
- Úgy véltük, a szoftverkérdés megoldását a tanárok által fejlesztett, mindenki számára hozzáférhető programok jelentik majd. Erre a válasz a kaliforniai SOFTSWAP és a minnesotai MECC cégek. Nem láttuk előre, hogy mennyire

nem egyszerű jó minőségű szoftvertermékeket előállítani és forgalmazni.

- Nagy jövőbeli jelentőséget tulajdonítottunk a jogvédelemnek. Tanárokból álló csoportok létrejöttét vártuk, amelyek komplett leckéket írnak majd, és ezeket lemezeken fogják a diákok számára hozzáférhetővé tenni. A jogvédelem magas költsége és a termékek gyenge minősége azonban nem tette vonzóvá ezt az eljárást.
- Két dolognak a megvalósulását sokkal korábbra jósoltuk. Az egyik a lemezkölcsönzés. Egyre népszerűbb és egyre választékosabb ugyan, a technikai problémák és a szoftvervédelem miatt azonban lelassult az intézményesítése. A másik a részletes magyarázattal ellátott adatközlő programok. Ez a rendszer a későbbiekben kényelmetlennek bizonyult: az útmutatókat nem vették figyelembe, ez a vonal is befulladt. Most a nagyobb teljesítményű gépektől, a kevésbé költséges hálózatoktól várjuk az információhordozó programok újraélédését.
- Megemlítem, hogy az utópisták a 70-es évek végén vezető helyre sorolták a videorendszereket, a video-lemezjátszókat. Számottevő darabszámban egyik sem áll még rendelkezésünkre, de el kell ismernünk, közeleg az áttörés ezen a területen.
- Két dologra egyáltalán nem számítottunk. Az egyik, hogy valamennyi iskolai évfolyamban általános lesz a szövegszerkesztők használata. De nem volt előre látható a többi alkalmazói szoftvernek tanterv szerinti használata sem. A Logo nem várt hatása a kicsikre és a tanárokra a másik örömteli meglepetés.

Sokban tévedtünk tehát. Viszont a számítógépes oktatás forradalma már nem látomás a távoli messzeségbe, hanem itt és most zajlik. Büszkék lehetünk arra, amit az oktatók a számítógépekkel megvalósítottak. Nagy előrelépést tükröz, hogy

— erős nemzetközi és helyi szervezetek jöttek létre szerte a világban. Kiadványaik, konferenciáik kiváló fórumai az elképzelések terjesztésének;

— az egyetemeken megindult a tanárképzés kezdő és haladó fokon egyaránt. A helyi központok meghatározó forrásai a képzésnek, a tananyagra vonatkozó speciális információknak, szoftvereknek stb.;

— az oktatóprogramok egyre jobbak, nagy kiadók is bekapcsolódtak a forgalmazásukba;

— országos és helyi szoftverismertetők nyújtanak segítséget a tanároknak a választáshoz az óriási kínálatból;

— mind egyetemi, mind helyi viszonylatban jelentős fejlődésnek indult a kutatás.

A jövőre utaló észrevételek

Elsőként a kompakt lézerlemezre gondolok. Jelenleg 1000 dollárba kerül az IBM PC-vel kompatibilis lemezmeghajtó és hozzá egy 21 kötetes lexikon kompakt lemezen. A lexikon mindössze egyötödét foglalja a lemeznek! A lemezt kezelő szoftver igen hatékony, a gyorsaság elképesztő. És mindez olyan áron, hogy felsőoktatási intézményeink könyvtárába már ma is könnyen telepíthetünk egy-egy ilyen egységet. De tekintsük a kompakt lemezt most egy szoftvertároló médiumnak. Képzeljenek el 1400 darab floppylemezt egy kupacban: 4 méter magas oszlopot tenne ki. Tartalmuk ugyanakkor ráfér egyetlen kompakt lemezre. Gondolják csak meg: az iskolájuk vagy iskolakörzetük egyetlen 5"-es lemezen kaphatná meg az összes szoftvert, amelyről aztán könnyen kiválogathatók lennének a sohasem használt és a nélkülözhetetlennek ítélt programok. (A kompakt lemezekkel egyébként részletesebben foglalkozik dr. Simonyi Endre cikke lapunk mostani számában, a 39. oldalon. — A szerk.)

És most a videóról. Az idei vancouveri EXPO-n új információközlő állomásokat mutattak be. Ezek valójában egy IBM PC-ből és egy új kereskedelmi termékből, az ún. IBM Infowindow Systemből álltak. Ezek az egységek feladatokat állítottak elő és küldtek a képernyőre. A feladatok, gyakorlatok számítógépes grafikából, fényképekből, szintetizált vagy élő beszédből és mozgó videóból álltak. Izgatottan várom, hogy milyen lehetőségeket hordoz ez az új rendszer a minőségi információközlés területén.

A jó minőségű hardver és szoftver azonban nem minden. A legfontosabb amit minden tanárnak személy szerint, egyedül kell véghezvinnie: keresni kell a megoldást, kipróbálni, újat és igényesebbet alkotni, egyszóval dolgozni azon, hogy miként lehet ennek a technikának az előnyeit valóban kihasználni. Hogy olyan iskolákban tanulhassanak az elkövetkező nemzedékek, amelyekben az eddigieknél jobban lehet tanulni és tanítani. A számítógépes oktatást favorizáló tanárok feszített tempóban igyekeznek is minél többet megtanulni a számítástechnikából. Újabb és újabb gépalkalmazásokat találnak ki az ismeretközlés javítására. A jövő iskolája valójában itt van már körülöttünk — részleteiben „fel van találva".

Néhány kísérletről

Az alaszkai Juneau-ban Paul Berg egyéves időtartamú programot szervezett számítógépet nem használó olyan tanárok részére, akikben megvan az akarat, hogy az írást-olvasást a megszokottól eltérően oktassák. A hallgatók fejlődése a kezdők újtól való félelmétől a megedződött felhasználók magabiztosságáig hű tükre legtöbbünk történetének. Paul Berg erről szóló jelentésének címe: Hand-in-hand: The Writing Process and the Microcomputer (Kéz a kézben: Az írás folyamata és a mikroszámítógép).

Tabiona kis falu Utah állam kanyonrengetegében, az isten háta mögött. Az ottani diákok korábban nem kaptak idegen nyelvi oktatást, csak miután az egyetemek megemelték a felvételi szintet. Ekkor a polgármester a kérdés megoldására tervet dolgoztatott ki egy munkacsoporttal, amelyben az oktatási osztály, az IBM, egy videotechnikával foglalkozó társaság és az idegen nyelvi oktatás egy szakértője dolgozott. Jelenleg naponta sugározzák a tananyagot műholdon keresztül, és rendelkezésre áll egy beszédszintetizátorral ellátott IBM PC, hogy segítse az anyag jobb elsajátítását. Az eredmény: a legtöbb diák megduplázta a tempót a spanyol nyelv tanulásában.

Marilyn Pollock ötödikeseket tanít egy Chicago melletti kisvárosban. 15 számítógép működik az osztályában. Ő a kedvenc példám a jó tradicionális tanártípusra. A rendelkezésére álló valamennyi erőforrást felhasználja, beleértve a férjét is, hogy ismereteket szerezzen az új technikáról, és hogy a technikát az oktatás céljainak szolgálatába állítsa. Marilyn osztálya felfedezi a tudományokat, társadalmi ismereteket szerez adatközlő programok segítségével, megtanul lemezre írni, és kapcsolatot tart távoli helyekkel a telekommunikáció segítségével. De a legfontosabb, hogy a gyerekek hozzászoknak, miként lehet és hogyan kell tanulni számítógépekben gazdag környezetben.

New Mexico egyik iskolájában megtalálták a módját, hogyan tehetnek szert olcsón új számítógépekre: az iskola elektronikai osztálya építi a gépeket, majd átadja a programozással foglalkozó osztályoknak.

Oaklandben (Kalifornia) a Franklin bentlakásos iskolában az öt-hat évesek, akik több mint húsz nyelvet beszélő közösségekből kerültek ki, négy számítógépet használnak az olvasóteremben. A számítógépes gyakorlás kiegészül kazettára listázással, beolvasással, rajzolással, gépeléssel, levelek kinyomtatásával. Ezek a gyerekek is az IBM Writing to Read programot gya-

korolják, amellyel már jelentős sikereket értek el az ország több mint 2000 iskolájában, az írás-olvasás oktatásában. Számukra a számítógépben semmi misztikus nincs: része a szoba berendezésének.

Edward Reese atya a San Jose-i (Kalifornia) katolikus iskola igazgatója. Amikor az egyik szülő felajánlotta, hogy kiépít egy terminálos rendszert az egyik osztályban, Reese atyának máris kész terve volt a gépek üzemeltetéséhez. Az egyes munkahelyeken alkalmazói szoftvereket futtatnak. Törölte a programozást a 9. osztályosok tantervéből, és szövegszerkesztők, táblázók (spread sheet), adatbázis-kezelők oktatását vette fel. Elsőként sajátította el a programok kezelését, majd az egyik szekciót oktatta.

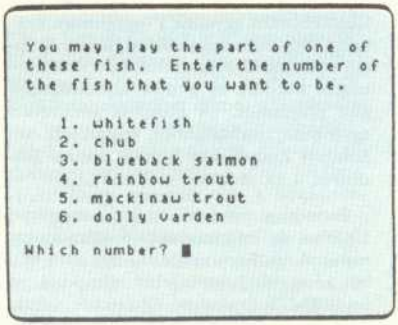
Denver belvárosában van a George Washington iskola. A város iskolái közül ez a legnépesebb, és neki van országos viszonylatban a legkiterjedtebb számítógépes oktatási programja. Ezt a programot egy matematikatanár dolgozta ki. 750 diáknak 27 szekcióban 15 tantárgyat oktatnak. Túl az ügyvitel, a matematika, a nyolcféle programozási nyelv oktatásán, van egy olyan kurzusuk, ahol diák, tanár egyaránt számítógépet szervizel és szoftvert készít a többi osztály számára. Egy diákcsoport a közelmúltban olyan programot fejlesztett ki, amely az angol nyelvet oktatja második nyelvként, és amely a laoszi ábécét is megjeleníti. Ez a szoftver a kereskedelemben is kapható.

Az itt következő, utolsó példámmal a mikrogépek nagy teljesítőképességére kívánok rámutatni, amely önmagában is új lehetőségeket kínál az oktatásügy minden területén.

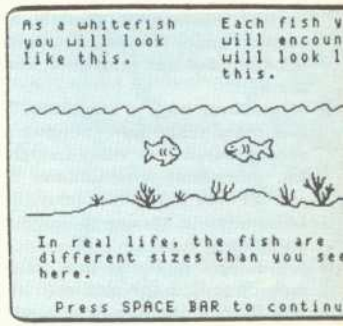
Kaliforniában van egy kies, mezőgazdasággal foglalkozó település, mindössze egy iskolával. Az iskolának rendkívül érdekes a tanmenete. A napi tananyagot az osztálytermekben felállított tévékészülékek sugározzák. A programozással foglalkozó osztályokon kívül van két különleges osztály. Az egyik a számtanfeladatok megoldásában nyújt segítséget a Multiplan programmal, a másik szövegszerkesztési feladatokat lát el. Rám a legnagyobb benyomást az angol nyelvi csoport tette, amely az elsősök nyelvi problémáit hivatott megoldani. A nehezen boldoguló diákok két sorozatból álló nyelvi kurzuson vesznek részt. Ezen a tanfolyamon a tananyagot hét kategóriába sorolták: olvasás, írás, helyesírás, nyelvtan, beszélgetés, előadások tartása, előadások hallgatása.

Ha szétnézünk magunk körül, láthatjuk, hogy az Államok legeldugottabb részein, a legszegényebb rétegekből származó gyerekek is szövegszerkesztőket használnak gondolataik megfogalmazására. Aztán rádöbbenünk arra is, hogy azok az IBM PCjr-ok, amelyeket ezek a gyerekek használnak, legalább olyan teljesítményűek, mint egy nagyszámítógép volt a hetvenes évek elején. Felemelő gondolat, hogy a számítógépes oktatás hívei, a tanárok, mint Önök vagy én, a magas szintű technikát a legjobb cél szolgálatába állították. Olyan oktatási segédeszközt hoztak létre, amit egy mérnök sem tud kigondolni.

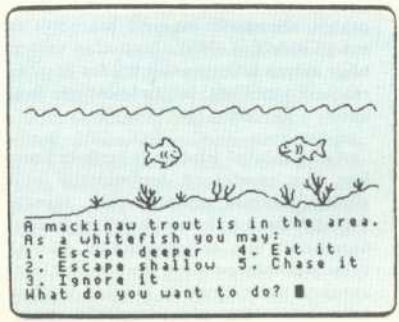
Mi, akik igyekszünk, de kénytelenek is vagyunk szakmailag továbbfejlődni, álmodozunk és alkotunk. Olyanok vagyunk, mint egy angoltanár egy kis kaliforniai faluban. Vannak diákok, akiket segítünk a megfelelő szintre feljutni, másoknak feltárjuk a lehetőségeket. És tudnunk kell, hogy a technika ki nem aknázott lehetőségei bennünk is rejlenek. Van-e ennél izgalmasabb?



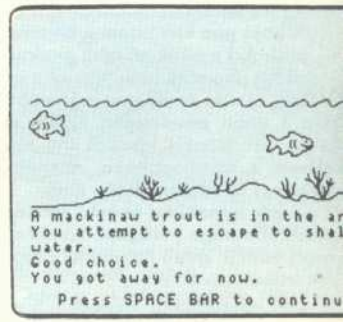
1. kép



2. kép



3. kép



4. kép

Utószó

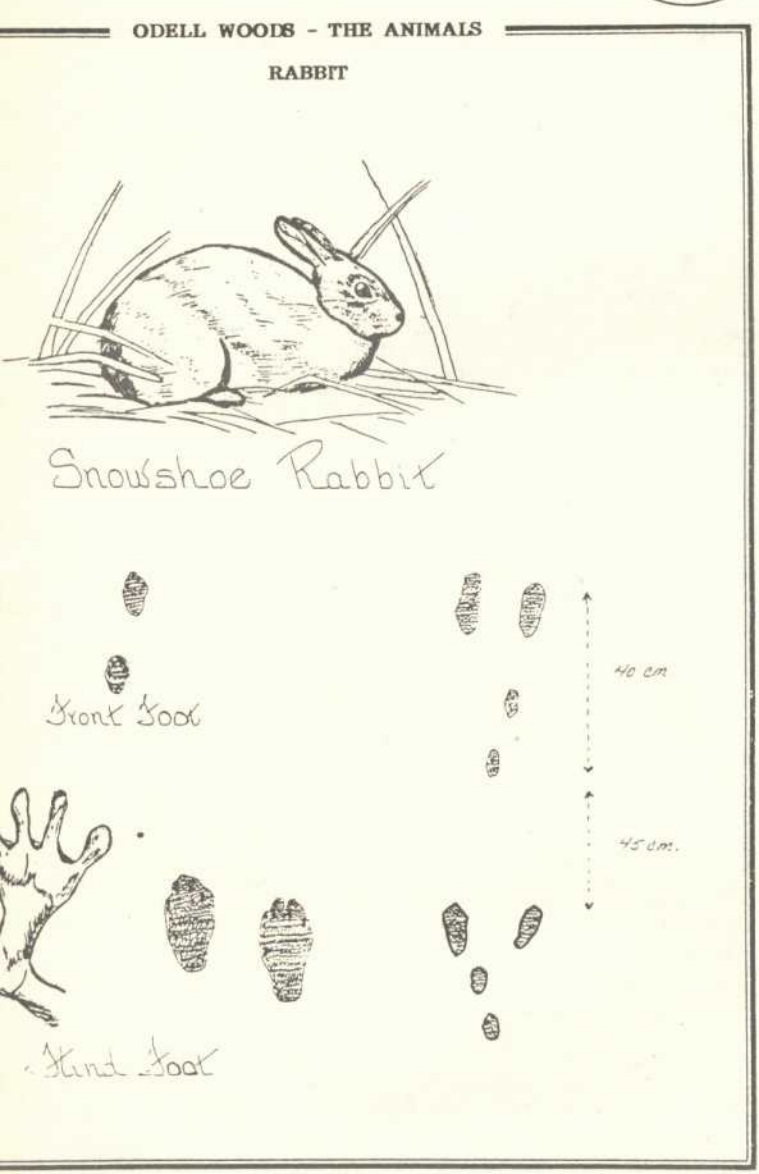
Lássuk, hogy körülbelül milyen is az a szoftver, amely kimondottan oktatási célra kerül forgalomba az USA-ban. Amint Wagner úr anyagából megtudhattuk, két nagy cég, a minnesotai MECC és a kaliforniai SOFTSWAP ezeknek a szoftvereknek az elsődleges előállítója. A MECC cég (Minnesota Educational Computing Consortium) nemzetközileg elismerten a vezető helyre tehető az oktatóprogramok gyártói között. 12 éves múltra tekint vissza; az akkori kétfős stáb mára 128 alkalmazottal működő vállalattá fejlődött. 1985-ben választékuk meghaladta a 300-at, és jóval több mint egymillió példányszámban volt forgalomban szerte a világban. A programok mintegy 60 százalékát Apple gépre írják, minthogy az egyesült államokbeli iskolákban ez a legelterjedtebb géptípus.

Ennek a cégnek a kínálatából mutatunk be egy programot, melyet Simonyi Endre a gyártótól az ismertetés engedélyével kapott.

A „Természet biológiája" tulajdonképpen két programból áll: az *Odell tó* a 3—9. osztályosoknak, az *Odell erdő* a 2—8. osztályosoknak készült. Az Odell tó programot az Oregon állambeli, azonos nevű tó adatai alapján állították össze. A programnak egyszerű, de vonzó grafikája van. Az Odell erdő, amely a tavat körülölelő erdő állatvilágának életét szimulálja, grafikával nem rendelkezik.

A „Természet biológiája" a táplálékláncok ökológiai összefüggését tanítja. A diák ki-

10c



5. kép

választja, hogy milyen állat szerepét kívánja játszani *(1. kép)*. Ha például vadnyúl lesz, és az erdőben más állatokkal találkozik, mondjuk egy éhes farkascsordával, választania kell, hogy elkergeti-e, megeszi-e őket, rájuk se hederít, vagy esetleg elszalad. Jelen esetben a futás a legjobb választás. Kilencféle találkozásra ad alkalmat a program egy futásban, és a diáknak a túlélésre kell játszania.

Ez egyszerűen hangzik, de nem mindig annyira magától értetődő a megoldás, mint az említett esetben. A növények, növényevők és húsevők kapcsolata bizonyos esetekben sokkal összetettebb, mint ahogy várjuk — kivált a tó táplálékláncában *(2., 3. és 4. kép)*. És a program még csak nem is tartalmazza a tápláléklánc valamennyi elemét!

Mindenesetre az embernek az a benyomása, hogy a program a tudásanyag észrevétlen fokozását szolgálja. Nem azt kívánja a gyerekekbe sulykolni, hogy nagyon járatlanok még a témakörben, de mégis sokszor le kell futtatniuk a programot ahhoz, hogy egy kis sikerélményben legyen részük. Sok magától értetődő dologhoz hozzátesz egy kevés nem kézenfekvőt, így a memorizálás nem követel különösebb szellemi erőfeszítést.

Mindkét programban beépített rutin szabályozza a diák által választott állattal szemben állók reakcióját. Például ha egy farkas találkozik egy farmerrel, a farmer rálő az állatra. A farmer találatainak aránya, ha a farkas nem ugrik a farmernek, hanem futásnak ered: 75%-ban téveszt, 25%-ban megsebesíti, 15%-ban megöli az állatot. Egy másik példában: kevesebb esélye van egy rókának a medve elől elszaladni, ha rossz erőnléti állapotban van. Így tehát hibázik, ha a nyúl elől elszalad (bár a találkozást így is túléli), vagy ha elkerüli a figyelmét, azaz ha nem eszi meg.

A „Természet biológiája"-t kitűnő kézikönyvvel látták el. Olvashatósága a megjelölt korosztályhoz igazodik. A 47 oldalas kézikönyvből húsz oldal a témával foglalkozó tananyag: leírja a tó kialakulását, ismerteti a terület állatvilágát. Az ábrák nemcsak az állat képét, hanem lábnyomát, sőt annak méreteit és járás közben kialakuló formáját is tartalmazzák *(5. kép)*. A kézikönyv többi része a programot ismerteti, és útmutatást ad a tanárnak a tárggyal foglalkozó biológiaóra megtartásához. Szép számban találhatók a könyvben személyi adatokkal együtt kitöltendő tesztlapok is.

VÁGH ISTVÁN

Szerkesztői megjegyzés. A cikk nyomdai előkészítése közben kaptuk a hírt, hogy a Commodore cég szerződést kötött a MECC céggel az Odell Lake (Odell tó) elnevezésű programnak a Commodore 64C géppel együtt szállított programként történő forgalmazására. Ezenkívül a géppel együtt szállítják még az ugyanebben a lapszámunkban ismertetett Geost is, valamint egy távadatátvitelt vezérlő programot.

PRIMO

Másolóprogram

Az alábbi, saját készítésű programommal könnyen lehet BASIC és gépi kódú programokat másolni.

Betöltés után a program kiírja parancsait, amelyeket a kezdőbetűkkel választhatunk ki. A másolandó program betöltése után kiírja az indítási címet, amennyiben automatikusan induló gépi kódú program volt az. Bármelyik gomb érintésével visszatérhetünk a menühöz.

A másolóprogramot a BASIC programmal lehet felírni magnóra. Ha ez megvan, LOAD paranccsal töltsük be a másolóprogramot, amely automatikusan indul.

A program az eredeti LOAD és SAVE utasításokat átírja, ezért ha már nincs szükség a programra, a tápfeszültség ki-be kapcsolásával hidegindítást kell végezni.

JUHÁSZ ISTVÁN

```
0 '**** Program: Juhasz Istvan ****
1 '**** Print   : HORZOL 86.10. ****
2 '**** Nev     : COPY betolto  ****
3 '#####################################
10 CLS
20 PRINT$6,10,"Egy Kis turelmet !"
30 FORI=22996TO23517:READA:POKEI,A:NEXT
35 CLS
40 PRINT$2,5,"A magnot allisd felvetelre!"
42 PRINT$4,5,"Ha kesz inditsd a magnot,"
44 PRINT"      majd nyomj meg egy billentyu
t !"
46 IFINKEY$=""THEN46ELSEPRINT$9,17,CHR$(2)
"OKE"CHR$(18):PRINTCALL(23414)
49 DATA0,0,0,0
50 DATA33,15,69,205,117,43,205,37,0,254,76
51 DATA40,26,254,108,40,22,254,83,202,143
52 DATA68,254,115,202,143,68,254,66,202,25
53 DATA26,254,98,202,25,26,24,223,33,124
54 DATA69,205,117,43,33,92,64,54,131,33,1
55 DATA0,205,143,0,205,117,60,33
60 DATA140,69,119,87,205,158,0,35,119,205
61 DATA158,0,35,119,205,158,0,35,119,205
62 DATA158,0,35,119,71,205,158,0,35,119,16
63 DATA249,205,158,0,35,119,205,117,60,35
64 DATA254,177,40,6,254,185,40,2,24,207
65 DATA119,205,158,0,35,119,205,158
70 DATA0,35,119,205,158,0,35,119,205,158,0
71 DATA35,119,43,43,43,43,126,254,185,194
72 DATA238,67,35,35,94,35,86,235,205,175
73 DATA15,205,102,56,254,0,40,249,195,238
74 DATA67,33,92,64,54,131,33,98,64,34,13
75 DATA69,33,12,69,54,16,205,146,0
80 DATA33,140,69,126,254,177,40,47,254,185
81 DATA40,57,205,221,58,126,205,244,58,35
82 DATA126,205,244,58,35,126,205,244,58,35
83 DATA126,205,244,58,35,70,126,205,244,58
84 DATA35,126,205,244,58,35,16,249,126,205
85 DATA244,58,35,24,204,50,92
90 DATA64,35,126,50,94,64,205,155,0,195
91 DATA238,67,245,205,221,58,241,50,92,64
92 DATA205,244,58,35,126,50,94,64,205,244
93 DATA58,35,126,205,244,58,35,126,205,244
94 DATA58,35,126,205,244,58,195,238,67,0,0
95 DATA0,12,2,32,32,32,32,42,42,42
100 DATA42,42,42,42,42,42,42,42,42,42,13
101 DATA32,32,32,32,42,32,32,66,89,46,32
102 DATA74,46,73,46,32,42,13,32,32,42,42
103 DATA42,32,32,67,32,79,32,80,32,89,32
104 DATA32,42,42,42,18,13,13,13,9,9,32,4
105 DATA76,20,32,79,32,65,32,68,13,13,9,9
106 DATA32
110 DATA4,83,20,32,65,32,86,32,69,13,13,9
111 DATA9,32,4,66,20,32,65,32,83,32,73,32
112 DATA67,13,13,9,9,32,0,73,78,68,73,84
113 DATA93,83,73,32,67,73,77,32,58,32,0
120 DATA33,212,89,17,234,67,1,166,1,237
121 DATA176,33,92,64,54,131,33,205,91,34
122 DATA203,91,33,202,91,54,16,205,146,0
123 DATA33,92,64,54,249,17,238,67,33,146
124 DATA69,1,0,0,205,149,0,22,0,205,221,58
125 DATA62,185,205,244,58,58,94,64,60,205
130 DATA244,58,62,238,205,244,58,62,67,205
131 DATA244,58,62,52,205,244,58,33,25,26
132 DATA229,201,0,0,0,67,79,80,89,32,32,32
133 DATA32,32,32,32,32,32,32,32,0
```

PRIMO — primátus

Keveset írnak a Magazinban a Primóról! (Az októberi szuperakció után remélhetőleg többet . . .) Engedjenek meg néhány elfogult megjegyzést a gépről.

A SUPER BITLET PRIMO-kínjában olvastam: „. . . a Primo rövidítve is elfogad BASIC-szavakat, ha megfelelően rövidítjük". Ezt a feltételt paraszti fordításomban kódolásként értelmeztem. A szoftverfüzet tartalmazza a kulcsszókészlet kódjait, amelyeket a 128—225, ún. definiálható karakterek grafikus jelével hívhatunk elő. Mint tudjuk, a ↓ a vele egyidejűleg nyomott billentyű kódját 128-cal növeli, a CTRL pedig 64-gyel csökkenti. Alkalmazásánál a háromkezű emberek előnyben vannak, ugyanis többször kell egyidejűleg 4 billentyűt is nyomni. Söt a '84.2 verzió egyenesen zsonglőrködésre késztet, ugyanis a kódmódosító billentyűk a klaviatúra két átellenes pontján találhatók. A megjelenített szavakból az is kiderül, hogy ez a kis gép megfelel a szélesebb perifériák követelményeinek, a szoftveres továbbfejlesztésnek. A ROM például tartalmazza a DEF, FN, MERGE, CMD, CVI, CVS, CVD szavakat.

Néhány gyakran használt parancs, utasítás, függvény képzése a következő:

ABS	↓Y
AND	↓R
ASC	↓v
ATN	↓d
AUTO	↓7
BEEP	↓CTR Ö
CALL	↓A
CDBL	↓q
CHR$	↓w
CINT	↓o
CLEAR	↓8
CLOSE	↓&
CLS	↓CTR D
CONT	↓3
COS	↓a
CREATE	↓+
CSGN	↓p
DATA	↓←
DEFINT	↓→
DEFSGN	↓CTR Z
DEFSTR	↓CTR X
DELETE	↓6
DIM	↓CTR J
EDIT	↓CTR Á
ELSE	↓CTR U
END	↓CTR É
ERL	↓B
ERR	↓C
ERROR	↓í
EXP	↓é
FIX	↓r
FOR	↓break
FRE	↓Z
GOSUB	↓CTR Q
GOTO	↓return
IF	↓CTR O
INKEY$	↓I
INP	↓ó
INPUT	↓CTR I
INT	↓X
LEN	↓s
LEFT$	↓x
LIST	↓4
LOAD	↓'
LOG	↓ú
MID$	↓z
NEW	↓;

1. lista

```
10 CLS:PRINTTAB(5)"R A J Z O L O"
20 REM
30 REM
40 REM
50 REM
60 REM
70 REM
80 REM
90 REM SZABALYOK SORAI
100 FOR I=0 TO 1000:NEXTI
110 PRINT:PRINT"BARMELY BILLENTYU INDIT"
120 IF INKEY$="" THEN 120
130 CLS:PRINT"KEZDO KOORDINATAK:"
140 INPUT X,Y
150 IF X<0 OR X>255 OR Y<0 OR Y>191        THEN PRINT"HIBA!":GOTO 130
160 CLS:SET(X,Y):S=1
165 A$=INKEY$
170 IF A$="J" AND X<255 THEN X=X+1
180 IF A$="B" AND X>0 THEN X=X-1
190 IF A$="L" AND Y>0 THEN Y=Y-1
200 IF A$="F" AND Y<191 THEN Y=Y+1
210 IF A$="T" THEN 130
220 IF A$="V" AND S=1 THEN S=2:            GOTO 260
230 IF A$="V" AND S=2 THEN S=1
240 IF A$="S" THEN STOP
250 IF S=1 THEN SET(X,Y)
260 IF S=2 THEN RESET(X,Y)
270 GOTO 165
```

NEXT	↓CTR G
NOT	↓K
ON	↓!
OPEN	↓"
OR	↓S
OUT	↓space
PEEK	↓e
POINT	↓F
POKE	↓I
RANDOM	↓CTR F
READ	↓CTR K
RESET	↓CTR B
RESTORE	↓CTR P
RESUME	↓↑
RETURN	↓CTR R
RIGHT$	↓y
RND	↓Ü
RUN	↓CTR N
SAVE	↓—
SCREEN	↓.
SET	↓CTR C
SGN	↓W
SIN	↓b
SQR	↓Á
STEP	↓L
STOP	↓CTR T
STR$	↓t
STRING$	↓D
TAB(	↓<
TAN	↓c
TEST	↓)
THEN	↓J
TO	↓=
TROFF	↓CTR W
TRON	↓CTR V
USING	↓?
VAL	↓u
VARPTR	↓É

Hónapokig tartó tanulással lerövidíthető a gépelési idő. A programokon belüli karakterdefiniálás az előbbi szavak képzési módját nem befolyásolja.

A dokumentációellátás igen szegényes. Az ún. felhasználói kézikönyv kritikán aluli: hiányos, megjelenése előtt is elavult, szakmailag hibás. A hardvert és a szoftvert ismertető füzetek adnak ugyan komolyabb betekintést a Primo lelkivilágába, de az alapos ismerethez sok próbálkozás kell. Érvényesül a Murphy-törvény átirata: „Legjobban a károm kodás nyelvét ismerjük"! Jó tanácsom tehát a szalonképességért és az időrablások ellen küzdő gyakorlatlan primósoknak: mielőtt kikezdesz a Primóval, lépj házasságra a HT-vel!

Két kezdő színvonalú programjavaslatom van. Az egyik a „Rajzoló". A gépkönyv végén közölt ugyanilyen című program az ott leírt módon nem működik. Az *1. lista* a négyirányú rajzoló javított változata, tehát nem új. A másik az Élet és tudomány 1985-ös zsebiskolájának csirkevadászat c. feladványa, Primóra átírva *(2. lista)*. Ez sem új. A listák C64-en készültek, „aki" erősen töri a magyart.

Végül két észrevétel. 1. A Primo érti a gépi kódhívás USR-jét. Ezt semmilyen dokumentációban nem találtam, a ROM-ban sem. 2. Az egyik B—32-es gép hibás kazettatöltés utáni programindításkor „S Error" hibajelzést ad, és csak a RESET-re reagál. Nincs rá magyarázatom.

TAKÁCS DEZSŐ

2. lista

```
5 CLS:PRINT"CSIRKEVADASZAT"
6 PRINT"A SAS SPACE-SZEL VADASZIK"
7 PRINT"KILEPES 'BRK'-VAL"
10 S=1:Z=0:G=1
11 INPUT"A SAS SEBESSEGE(1-10)";V
12 IF V<1 OR V>10 THEN 11
15 CLS:U=(1+V)/2
20 FOR I=0 TO 255:SET(I,0):NEXT
55 C=20+RND(220)
60 FOR I=C-2 TO C+2:SET(I,2):NEXT
90 M=50+RND(100)
100 Y=M:Y1=M
105 FOR R=1 TO 255
110 IF U+R>255 THEN CLS:GOTO 20
115 RESET(U+(R-1),M):RESET(U+(R-1),M+1)
120 SET(U+R,M):SET(U+R,M+1)
130 IF INKEY$=" " THEN 170
135 FOR Q=0 TO 10:NEXT Q
145 NEXT R
150 FOR J=0 TO 1000:NEXT
155 CLS
160 GOTO 20
170 FOR I=1 TO 300
175 X=U+(R+1)
180 IF X>255 THEN 240
185 RESET(X-U,M):RESET(X-U,M+1)
190 RESET(X-U,Y):RESET(X-U,Y+1)
195 SET(X,Y)
200 Y2=2*Y1-Y-G
210 Y=Y1:Y1=Y2
215 IF Y<4 AND ABS(C-X)<3 THEN 260
220 IF Y<1 THEN 240
230 NEXT I
240 PRINT"NEM JO!":S.:"LECSAPASBOL":Z:"TALALT"
245 S=S+1
250 GOTO 150
260 Z=Z+1
265 PRINT"JO!":S.:"LECSAPASBOL":Z:"TALALT"
266 S=S+1
270 RESET(C-1,2):RESET(C+1,2)
280 GOTO 150
```

PRIMO

Gépi kódban

Értékadás

A G1 jelű BASIC programban *(1. lista)* lévő gépi kódú rész a HL jelű regiszterben elhelyezi a négy értéket: LD (HL),Ø4, és visszatér a BASIC programba: RET.

Az 1. sorban a program neve van.

A 1ØØ-as sorban a programban lévő skalár változóknak adunk értéket, mert ha a gépi kódú program helyének meghatározása után és működése előtt „új" skalár változót használunk, akkor a program hibásan működik.

A 11Ø-es sorban K% (index) részére helyet foglalunk. E helyre fogjuk elhelyezni a gépi kódú programot. A lefoglalt hely természetesen lehet nagyobb a szükségesnél, de kisebb nem.

A 12Ø-as sorban megkeressük, hogy a gépi kódú program részére lefoglalt hely hol van.

A 129-es sor csak magyarázó rész. Elhagyható.

A 13Ø-as sorban betöltjük a gépi kódú programot.

A 14Ø-es sorban meghívjuk a gépi kódú programot. A gép a HL jelű regiszter címét a C változóban adja meg.

A 15Ø-es sorban kíratjuk ezt a címet és az ott tárolt értéket.

A program működését kipróbálhatjuk Ø és 255 között lévő egész számokkal. A programot kipróbálás előtt rögzítsük szalagra, mert hiba esetén törölheti magát.

A tanulás kedvéért rontsuk el a programot a következő módokon:

— A HL regiszterbe tört számot töltünk.
— A HL regiszterbe negatív számot töltünk.
— A HL regiszterbe 255-nél nagyobb számot töltünk.
— A 12Ø-as és a 14Ø-es számú sorok között új skalár változónak adunk értéket. Például így:

135 FOR I=1 TO 1ØØ:NEXT I

Adatátadás

A G2 jelű BASIC programban *(2. lista)* lévő gépi kódú rész adatot kap a DE regiszteren keresztül, áthelyezi az A regiszterbe: LDA, (DE), majd áthelyezi a HL regiszterbe: LD(HL), A és visszatér a BASIC programba: RET. A gép sajátossága, hogy az adatot a DE regiszter fogadja, és a HL regiszter adja vissza.

A 11Ø-es sorban helyet foglalunk az adat számára is.

A 122-es sorban megkeressük, hogy az adat részére lefoglalt hely hol van.

A 132-es sorban a helyére rakjuk az adatot.

A 14Ø-es sorban úgy hívjuk meg a gépi kódú részt, hogy egyben átadjuk az adatot is.

Az 1., 1ØØ., 1Ø5., 12Ø., 129., 13Ø. és a 15Ø-es sorok a G1 jelű program után nem igényelnek magyarázatot.

Ezt a programot is próbáljuk meg elrontani a G1 programnál leírt módszerekkel!

SOMOGYI GYÖRGY

1. lista

```
1 REM G1
100 E=0 : CC=0
110 DIM K%(20)
120 CC=VARPTR(K%(0))
129 REM : LD(HL),04 : RET
130 POKECC,54,4,201
140 C=CALL(CC)
150 PRINT C,PEEK(C)
```

A PROGRAM	A KÓD
LD(HL),04	54,4
RET	201

2. lista

```
1 REM G2
100 C=0 : CC=0 : CD=0
105 INPUT V
110 DIM K%(20),L%(20)
120 CC=VARPTR(K%(0))
122 CD=VARPTR(L%(0))
129 REM : LDA,(DE) : LD(HL),A : RET
130 POKECC,26,119,201
132 POKECD,V
140 C=CALL(CC,CD)
150 PRINT C,PEEK(C)
```

A PROGNAM	A KÓD
LDA,(DE)	26
LD(HL),A	119
RET	201

Tv-foci

A program abban különbözik a korábbi változatoktól, hogy mivel a labda sebessége 1 és 5 között szabályozható, többféle kategóriában teszi lehetővé a játékot. A bal kapust az A és Y billentyűkkel, a jobb kapust az ó és ú billentyűkkel lehet irányítani.

```
 1 CLS
 5 K=3
10 PRINT$
   1,5,CHR$(3)CHR$(2);"***PRIMO-
   foci***"; CHR$(18)
20 PRINT$
   3,0,STRING$(41,45):PRINT$ 6,2,
   "BAL KAPUS":PRINT$6,28,"JOBB
   KAPUS"
30 PRINT$ 7,3,"FEL:A(a)":PRINT$8,3,
   "LE:Y(y)"
40 PRINT$ 7,29,"FEL:ó":PRINT$ 8,29,
   "LE:ú"
50 PRINT$ 14,8,"NYOMJ MEG EGY
   GOMBOT!!!"
60 IF INKEY$=""THEN 60
65 GOTO 500
70 CLS:PRINT CHR$(1):FOR I= 3 TO
   252: SET(I,1):SET(I,191):NEXT I
80 FOR I=1 TO 55:SET(3,I):SET(252,I):
   NEXT I
90 FOR I=143 TO 191:SET(3,I):SET
   (252,I):NEXT I
100 U=0
110 S=4:W=10
120 N=1
130 PRINT CHR$(6)
140 H=2
150 RANDOM
160 U=U+H
170 C=C+H
180 PRINT$ S,40,CHR$(138)
190 PRINT$ W,0,CHR$(138)
200 PRINT$ C1,U1," "
210 PRINT$ C,U,"o"
220 U1=U
230 C1=C
235 FOR I=0 TO M:NEXT I
240 IF (C=W AND U=0) OR (C=S
   AND U=40)THEN H=-H:GOTO
   160
250 IF U=>40 OR U<=0 THEN
   H=-H
260 IF C=>14 OR C<=0 THEN
   N=-N
270 F$=INKEY$
275 IF F$="A" AND NOT(W=4) THEN
   PRINT$ W,0," ";:W=W-1
280 IF F$="a"AND NOT(W=4) THEN
   PRINT$ W,0," ";:W=W-1
290 IF F$="Y" AND NOT(W=10)
   THEN PRINT$ W,0," ";:W=W+1
295 IF F$="y" AND NOT(W=10)
   THEN PRINT$ W,0," ";:W=W+1
300 IF F$="ó" AND NOT(S=4) THEN
   PRINT$ S,40," ";:S=S-1
310 IF F$="ú" AND NOT(S=10) THEN
   PRINT$ S,40," ";:S=S+1
320 IF C>=4 AND C<=10 AND (U=0
   OR U=40) THEN PRINT$ C,U,
   "";:GOTO 340
330 GOTO 160
340 IF U>20 THEN E=E+1 ELSE
   R=R+1
350 PRINT$ 0,0,E;":";R;CHR$(18)
360 IF E=10 OR R=10 THEN 410
370 IF INKEY$=""THEN 370 ELSE
   PRINT$ 1,13,"15 üres";
380 C=FIX(RND(0)*13+1):IF
   RND(0)>0.25 THEN H=2*SGN(H)
390 GOTO 160
400 CLS:PRINT$ 8,5,"REMÉLEM JÓL
   SZÓRAKOZTAK": GOTO 470
410 FOR I=0 TO 1000:NEXT I
420 CLS:PRINT"JÁTSZANAK MÉG?(I/
   N)"
430 F$=INKEY$:IF F$=""THEN 430
440 IF F$="N" THEN 400
450 IF F$="I" THEN RUN 70
470 GOTO 470
490 PRINT CHR$(12):PRINT$ 8,7,CHR
   $(2) "!!! 1...5!!!"CHR$(18):PRINT:
   PRINT:PRINT:PRINT
500 INPUT"MILYEN KATEGÓRIÁ-
   BAN JÁTSZANAK? (1...5)";M
510 IF M<1 OR M>5 THEN 490
520 IF M=2 THEN M=10
530 IF M=3 THEN M=20
540 IF M=4 THEN M=40
550 IF M=5 THEN M=60
560 GOTO 70
```

HORVÁTH JÁNOS

A program kérésre kiírja az információkat; ezután a változók értékadása, majd a pálya kirajzolása következik. Gombnyomásra elindul a játék. Ketten játszhatják; az egyik játékos ↑ és ↓ nyilakkal, a másik az O és L betűkkel tudja mozgatni „embereit". Mindkét játékosnak két-két „embere", azaz ütője van: egy a kapujában, egy pedig az ellenfél térfelének közepén. A két ütőt egyszerre lehet mozgatni. Ha a labda falnak vagy ütőnek ütközik, akkor rugalmasan visszapattan, ha bekerül az egyik kapuba, akkor a legalsó sorban kiírt állás megváltozik: a gólt kapó fél pontja eggyel nő. Ha valamelyik játékos 10 gólt kapott, vége a játéknak, és a gép kiírja a győztest. Ha folytatódik a játék, tehát még nincs 10 gól, akkor a labda a gólt kapó játékos kapujától kerül játékba, mintha a kapus kirúgná.

Esetleg magyarázatra szorul a 390-es és az azt követő sorok szerepe. Miután a program kiszámolta az új koordinátákat (a labdáét), megvizsgálja, nincs-e ott ütő vagy fal. Ha igen, akkor a régi koordinátákat számolja ki, a fal vagy ütő elé kirajzolja a labdát, majd egy kis idő múlva leoltja, és az ellentétes irányban halad tovább a labda (ütközés). Az 560-as sor a gól után a labda irányát ellentétesre állítja, így az újra játékba kerül.

A labda úgy mozog, hogy az X és Y koordinátáihoz hozzáadjuk U és V változó értékét (2 vagy −2). Ha értékük ellentétes lesz, akkor megfordul a labda útja.

A program magyarázata

10—170	Információ.
200—299	A változók értékének megadása és a pálya kirajzolása.
300—345	A billentyűzet figyelése, a labda és a játékosok kirajzolása.
350	A labda új helyének kiszámítása.
360—370	Megvizsgálja, nincs-e gól.
380—430	Megvizsgálja, hogy ütközik-e falnak vagy ütőnek a labda.
500—560	A gólokkal foglalkozó rész.
520—540	Kiírja az állást: ha van 10 kapott gól, akkor vége a játéknak.
560—	A gólt kapó játékos kapuja felől kerül játékba a labda.

```
10 CLS
20 PRINT"FE'L KE'PERNYO !":PRINT
30 INPUT"* KERSZ INFORMACIOT * ";A$
40 IF LEFT$(A$,1)<>"I",200 ELSE CLS
50 PRINT"* * * * * * * * * * * * *"
60 PRINT"*                       *"
70 PRINT"*  T ' -  J A T E K   *"
80 PRINT"*                       *"
90 PRINT"* * * * * * * * * * * * *"
100 PRINT:PRINT" 1. JATEKOS: FEL ES LE A"
110 PRINT"              NYILAKKAL MOZOGHAT."
120 PRINT" 2. JATEKOS: O - FEL"
130 PRINT"             L - LE"
140 PRINT:PRINT"AZ VESZIT, AKINEK ELOBB LESZ"
150 PRINT"TIZ KAPOTT GOLJA."
160 PRINT:PRINT@MEHET A JATEK ?"
170 IF INKEY$="",170
200 CLS:DEFINT A-Z:A=15810
210 B=15837:X=32:Y=23
220 U=2:V=2:Z=14338:H=14400:C=0
230 FORI=0 TO 62:SET(I,2):SET(I,42):NEXTI
240 FORI=2 TO 9:SET(0,I):SET(62,I):NEXTI
250 FORI=35 TO 42:SET(0,I):SET(62,I):NEXTI
260 FORI=2 TO 42 STEP2:SET(31,I):NEXTI
270 PRINT@960,"HA MEHET, NYOMJON LE VALAMIT";
280 IF INKEY$="",280
290 PRINT@960,"AZ ALLAS:                              ";
299 REM * FOPROGRAM *
300 SET(X,Y):K=PEEK(H):L=PEEK(Z)
310 IFK=8 AND A>15488,POKEA,32:POKEA+20,32:A=A-64:GOTO330
320 IFK=16 ANDA<16180,POKEA,32:POKEA+20,32:A=A+64
330 IFL=128 ANDB>15488,POKEB,32:POKEB-20,32:B=B-64:GOTO345
340 IFL=16 ANDB<16200,POKEB,32:POKEB-20,32:B=B+64
345 POKEA,149:POKEB,149:POKEA+20,149:POKEB-20,149
350 RESET(X,Y):X=X+U:Y=Y+V
360 IF X=0 AND Y>9 AND Y<35,500
370 IF X=62 AND Y>9 AND Y<35,510
380 IF Y=3 OR Y=41,V=-V
390 IF POINT(X,Y),X=X-U:Y=Y-V ELSE 300
400 SET(X+U/2,Y+V/2):FORI=1 TO 20:NEXTI
420 RESET(X+U/2,Y+V/2):Y=Y+V:U=-U
430 GOTO 300
500 B1=B1+1:GOTO520
510 B2=B2+1
520 PRINT@970,B1;"    -    ";B2;
530 C=C+(B2=10)+(B1=10)*2
540 IF C,CLS:PRINT@512,-C;"-ES JATEKOS NYERT !";:END
560 X=(SGN(-U))*6+X:U=-U:GOTO 300
```

A változók szerepe

A	Az 1-es csapat kapusának a helye.
A+20	Az 1-es csapat játékosának a helye.
B	A 2-es csapat kapusának helye.
B—20	A 2-es csapat játékosának helye.
C	A győztes csapat (játékos) száma.
B1	Az 1-es csapat kapott góljainak száma.
B2	A 2-es csapat kapott góljainak száma.
X, Y	A labda jelenlegi X és Y koordinátája.
Z, H	Értékük 14 338 és 14 400; a billentyűzet figyelését könnyítik meg.
U, V	Minden alkalommal ezek értékével növeljük meg X és Y értékét (értékük 2 vagy —2). Ellentettjük használatakor a labda ellentétes irányba mozog.
I	Ciklusváltozó.
K, L	A billentyűzet figyelése alatt értékük PEEK(Z) és PEEK(H) (Z és H értékét lásd feljebb!).

BACZÓ TAMÁS

Z80 programozási gyakorlatok 7.

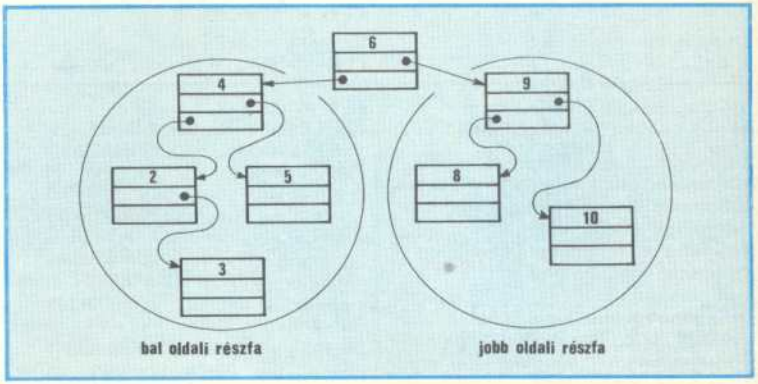
Cikksorozatom utolsó részében egy nagyon sokat vitatott problémáról, a rekurzióról szólok, és ennek kapcsán egy kicsit túl is tekintünk az assembly programozás keretein.

Sokszor hangoztatott vélemény, hogy a rekurzió mint programozási fogás felesleges, mert amit rekurzióval meg lehet írni, azt meg lehet anélkül is. Azt mondják, gépi kódban nem készíthető rekurzív program, és mégis minden megoldható ezen a nyelven. Nos, ez az érvelés nem egészen jó, mert van, amit nem érdemes megírni rekurzió nélkül, és mert gépi kódban sem lehetetlen rekurzív programok készítése, csak elég nehézkes.

Nézzünk meg egy rendkívül egyszerű, bináris fát kiíró szubrutint! A felépített fára az *ábra* mutat példát. Ennek a fának minden egyes eleme tartalmaz valamilyen értéket, továbbá két mutatót, amelyek az elem bal, illetve jobb fáira mutatnak. A fa felépítése olyan, hogy a gyökér bal oldali részfájának minden eleme kisebb értéket hordoz, mint a gyökér, a jobb oldali pedig nagyobbat. Ez igaz a részfák gyökereire és azok mindkét oldali részfáira, és így tovább. A kiíratás úgy megy végbe, hogy az elemek értékei nagyság szerint következzenek. Világos, hogy mielőtt a gyökérhez tartozó értéket kiíratnánk, ki kell íratni annak bal oldali részfáját, hiszen ott minden elem értéke kisebb a gyökér értékénél. A jobb oldali részfa viszont csak a gyökér után következhet, hiszen ott minden érték nagyobb a gyökérénél. A kiíratás tehát valahogy így megy:

1. Írassuk ki a gyökér bal oldali részfáját.
2. Írassuk ki a gyökér értékét.
3. Írassuk ki a gyökér jobb oldali részfáját.

Ezzel meg is van a szubrutin. Csupán az 1. és 3. pontokkal van gond: hogyan írassuk ki a részfákat? Nem másként, mint a most megírt szubrutinnal. Ez a rekurzív hatás. Nagyon fontos, hogy valahol kilépjünk a rekurzióból, azaz ha nincs bal vagy jobb oldali részfa, akkor ne hívja meg a szubrutin önmagát a nem létező részfa kiíratására.



Az eljárást PASCAL nyelven mutatja az *1. program;* a nyelv támogatja a rekurzív hívást.

Nézzük meg, hogyan lehet megírni a szubrutint gépi nyelven! A PASCAL függvény argumentuma egy mutató. A Z80 szubrutin *(2. program)* ezt a mutatót az IX regiszterben várja, azaz mielőtt valamilyen program meghívja a WRITE szubrutint, előtte el kell helyeznie IX-ben a gyökér címét. A szubrutin megvizsgálja, hogy van-e bal oldali részfa, és ha nincs, akkor a NOKIR címkénél — ahol a gyökér értékét írja ki a program — folytatódik a futás, majd megnézi, hogy van-e jobb oldali részfa. Ha nincs, akkor a szubrutin befejezte a működését, és visszatér.

Lássuk, mi történik akkor, ha van bal oldali részfa! Ebben az esetben a szubrutin meghívja önmagát, de előtte az IX regiszterbe helyezi a bal oldali részfa gyökerének címét. A visszatérés után ki kell íratni a gyökér értékét, de ehhez szükség van annak címére, ezért meg kell őrizni IX értékét a rekurzív hívás előtt. Nem használhatunk valamilyen változót erre a műveletre, hiszen akkor azt is el kellene menteni a hívás előtt, mivel a szubrutinnak szüksége van rá. Olyan helyre kell IX-et elmenteni, ahol biztonságban van, azaz amelyik helyet a szubrutin nem érinti. Ez a hely a verem. Amikor a jobb oldali részfát íratjuk ki, akkor már nem rekurzív módon hívja meg önmagát a program. Természetesen felesleges lenne

1. program

```
AE13   20 PROGRAM BINTREE;
AE13   30 TYPE TREE=RECORD
AE13   40       NUM:INTEGER;
AE13   50       LEFT,RIGHT:^TREE
AE13   60     END;
AE13   70       POINTER=^TREE;
AE13   80 {$L-}
AE13  100 PROCEDURE WRITREE(P:POINTER);
AE1F  110 BEGIN
AE37  120
AE37  130  IF P^.LEFT<>NIL THEN
AE52  140                      WRITREE(P^.LEFT);
AE68  150
AE68  160  WRITELN(P^.NUM);
AE78  170  IF P^.RIGHT<>NIL THEN
AE98  180                      WRITREE(P^.RIGHT)
AEA7  190 END;
AEB7  200 {$L-}
End Address: AEC2
```

2. program

```
FEEE                    ORG     100H
                        ;*************************************
                        ;*      2. PROGRAM                   *
                        ;*          Binaris fat listazo      *
                        ;*                  szubrutin        *
                        ;* IX-be ertek                      *
                        ;*      bal                         *
                        ;*      jobb                        *
                        ;*                                  *
                        ;*          V.P. 7C2.Nov.           *
                        ;*************************************
0100 DD6ED2     WRITREE LD      L,(IX+2) ;HL:=
0103 DD66D3             LD      H,(IX+3) ;     left
0106 7C                 LD      A,H
0107 B5                 OR      L
0108 2808               JR      Z,NOKIR  ;ha left=NIL
010A DDE5               PUSH    IX       ;IY elmentese---+
010C E5                 PUSH    HL       ;IX:=           !
010D DDE1               POP     IX       ;     HL        !
010F CD0001             CALL    WRITREE  ;rekurziv hivas !
0112 DDE1               POP     IX       ;IX vissza------+
0114 DD6E00     NOKIR   LD      L,(IX+0) ;HL:=
0117 DD6601             LD      H,(IX+1) ;     num
011A CDEEFE             CALL    WRITHL   ;HL kiirasa
011D DD6E04             LD      L,(IX+4) ;HL:=
0120 DD6605             LD      H,(IX+5) ;     right
0123 7C                 LD      A,H
0124 B5                 OR      L
0125 C8                 RET     Z        ;ha right=NIL
0126 E5                 PUSH    HL       ;IX:=
0127 DDE1               POP     IX       ;     HL
0129 18D5               JR      WRITREE  ;ciklus!!!!

Workarea - A3E5 to A526
ORG  end - 0128
LOAD end - 0000
```

elmenteni a mutató értékét, rá már nincs többé szükség. Ez a hívás megfelel egy egyszerű ciklusnak. Ciklus helyett nem szabad rekurziót alkalmazni!

Gyakori, hogy a rekurzió demonstrálására a faktoriális-számító programot hozzák fel példának, rendkívüli egyszerűsége miatt. Programozástechnikai szempontból azonban ez éppen a legtipikusabb ellenpélda arra, hogy mikor nem szabad rekurziót alkalmazni!

Térjünk vissza a bináris fa kiíratásához! Mennyivel egyszerűbb mindez PASCAL nyelven! Nem kell az előbbiekkel törődni, a nyelv mindezt megteszi helyettünk.

Vajon a változóit mind belenyomja a verembe? Természetes, hogy nem tárolhat mindent regiszterben; ha sok változó van a függvényben, akkor a többségét memóriában kell tárolni. Ugyanúgy, ahogy hosszabb assembly program megírásakor teszszük: helyet foglalunk például DEFS direktívával. Az ilyen változók a program futása során mindig ugyanott vannak, tehát egy rekurzív hívás előtt el kell őket menteni. Mivel a fordító nem tudhatja egy hívásról, hogy rekurzív-e, hiszen lehet, hogy A függvény hívja B-t és B hívja A-t, ezért minden függvényhívás előtt el kell menteni az összes változót.

Ennél egyszerűbb, ha a függvény változóit dinamikusan helyezzük el, azaz amikor egy függvényt meghív a fő programrész vagy egy másik függvény, akkor a változók számára szükséges tárhelyet megcímezzük egy mutatóval, és az egyes változók címét a mutató plusz egy eltolási érték adja. Z80 assemblynél ez a mutató lehet az IX vagy IY regiszter, 6502 processzornál egy nullalapos tároló, 8086 processzornál a BP regiszter.

Amikor a függvény meghív egy másik függvényt, akkor a mutató régi értékét elmenti a verembe, majd beállítja az új értéket, és végrehajtja a hívást. A visszatérés után POP utasítással kerül vissza a mutatóba a régi érték, és folytatódik a függvény végrehajtása. A fő programrész változóit nem dinamikusan tárolják; ezek ugyanolyanok, mint assemblyben a DEFS direktívával lefoglalt területeken levő változók; tehát fix címük van a futás során, míg a függvények egyéb változóinak a címe más és más attól függően, hogy honnan és mikor lett meghíva a függvény.

Programunkban volt még egy lépés, amelyet — bár a feladat egyszerűsége miatt az előbb nagyon könnyen megoldottunk —, részletesebben meg kellene vizsgálnunk. Ez a paraméterátadás. Nagyon hasonló probléma, mint a helyfoglalás a függvény saját változói számára. A különbség csak annyi, hogy ezeknek a változóknak kezdőértékük van.

Maradjunk a PASCAL nyelvnél, és csak az értékkel hívott paraméterekkel foglalkozzunk! A paraméterek helyét tehát ugyanúgy le kell foglalni, de még a függvény meghívása előtt ezekre a helyekre az aktuális paramétereknek megfelelő értéket el is kell helyezni. Szokás, hogy mind a paraméterek, mind a lokális változók számára a helyet a veremben biztosítják a fordítók. Ennek az az előnye, hogy a paraméterértékek elhelyezhetők a PUSH utasítással, a helyfoglalás pedig a többi változó számára a veremmutató csökkentésével érhető el. A mutató maga az SP vagy valamely más megfelelő regiszter lehet, amelybe az SP-vel egyező értéket töltünk. Ez célszerű is, hiszen például egy értékadás is szubrutinhívások sorozatává válhat a fordítás során.

Az eddig leírtak természetesen nem tények, csupán szokások, amelyeket az ésszerűség diktált. Biztos például, hogy egy 6502 processzoros gépen futó PASCAL program nem használja a vermet paraméterátadásra, hiszen az első lap 256 bájtja erre nem alkalmas, de minden bizonnyal valamilyen hasonló struktúrával dolgozik. A ZX-Spectrumon futó PASCAL nyelv a leírt módon oldja meg a problémát, és az említett mutató az IX regiszter. Ez a processzor sokkal alkalmasabb az ilyesmi megvalósítására. A 8086 processzort a tervezői úgy konstruálták meg, hogy utasításkészlete egyenesen sugallja, hogy a paramétereket a veremben adjuk át.

Gyakorlatként ajánlom az olvasónak, hogy próbálja meg megírni — először esetleg magas szintű nyelven — Hoare 1962-ben kitalált gyorsrendezési algoritmusát. Az algoritmus megtalálható például Aho—Hopcroft—Ullman: Számítógép-algoritmusok tervezése és analízise c. könyvben (Műszaki Könyvkiadó, 1982).

VERHÁS PÉTER

BASIC és gépi kód

Legutóbb a számláló utasításokról és a közvetett JMP utasításról volt szó. Most az utasítások egy újabb csoportjával ismerkedünk meg. Megértésük a különleges címzési mód miatt egy kicsit nehezebb az eddigieknél.

Feltételes ugróutasítások

Az eddig megismert utasításokkal csak olyan gépi kódú rutinokat tudunk írni, amelyekben a végrehajtás sorrendje rögzített, és ettől a sorrendtől a futás során nem lehet eltérni. Az itt következő utasítások lehetővé teszik, hogy a végrehajtás sorrendje bizonyos feltételektől függően különböző legyen.

Nyolc utasítás tartozik ebbe a csoportba. Nevük az angol BRANCH (elágazás) szó kezdőbetűjéből és az ugrás feltételének kétbetűs angol rövidítéséből áll. A feltétel az állapotbájt egyes bitjeinek pillanatnyi értékéből adódik, ez a segédletben is megtalálható. A feltételbitek értékét az ugróutasítások nem módosítják.

A relatív címzési mód

A feltételes ugróutasítások mindig kétbájtosak, és a relatív címzési módot használják. Ez azt jelenti, hogy az ugrási feltétel teljesülése esetén az utasításszámláló regiszter (PC) tartalmához egybájtos előjeles számként hozzáadják az utasítás operandusát. A program futása az így keletkezett új címen folytatódik. Az operandus az ugrási címnek és az ugróutasítást követő utasítás címének előjeles különbségét tartalmazza. Gépi kódban programozva ezt az értéket nekünk kell kiszámítani; vigyázni kell, mert könnyű eltéveszteni! Velem is többször előfordult, hogy az ugróutasítástól számoltam a távolságot, nem az azt követőtől.

Könnyebb dolga van annak, aki valamilyen assemblerrel dolgozik. Az operandust kétbájtos abszolút címként kell megadnia; az egybájtos gépi kódú operandust az assembler kiszámítja.

A segédletben a relatív címzési mód leírásába hiba csúszott. Az első bekezdés pontos, az azt követő példában a BEQ $24 nem 34-gyel, hanem 36-tal viszi előre a PC-t, míg a BEQ $FF nem 126-tal, hanem csak 1-gyel hátra. (Ami ugyancsak furcsa eredményt produkál, tekintve, hogy az ugrás a BEQ utasítás operandusára történik.) A következő bekezdésben is hiba van, ugyanis hátra 128, előre 127 bájtnyit ugorhatunk.

Az utasítások leírása

BPL — Branch on PLus. Ugrás, ha az N-bit tartalma Ø.

BMI — Branch on MInus. Ugrás, ha az N-bit 1-et tartalmaz.

BVC — Branch on oVerflow Clear. Ugrás, ha a V-bit Ø-t tartalmaz.

BVS — Branch on oVerflow Set. Ugrás, ha a V-bit 1-et tartalmaz.

BCC — Branch on Carry Clear. Ugrás, ha a C-bit Ø-t tartalmaz.

BCS — Branch on Carry Set. Ugrás, ha a C-bit tartalma 1.

BNE — Branch Not Equal. Ugrás, ha a Z-bit tartalma Ø.

BEQ — Branch on EQual. Ugrás, ha a Z-bit 1-et tartalmaz.

A feltételes ugróutasítások alkalmazását legközelebb egy gyakorlati példán mutatom be.

Néhány észrevétel

Székely Béla kollégám figyelmeztetett, hogy az 1986. szeptemberi számban közölt ROM-cím összehasonlító táblázatban hiba van, a C16 oszlopban a negyedik cím helyesen $9314. Utánanéztem, a gépelési hibát én követtem el, elnézést kérek. A júniusi számban megjelent BASIC betöltőben a hibátlan cím szerepel.

Kollégám néhány nappal később megemlítette, hogy a kiszámított GOTO alprogramom kétszer olyan hosszú, mint szükséges. Igaza van, de erről majd máskor írok.

BARNA LÁSZLÓ

A programozás Trabantja

Semmiképpen nem véletlen, hogy a programozással kapcsolatban az emberek egyre növekvő többsége a BASIC-re gondol. Ugyanakkor a hivatásos programozók gyakran óva intenek e nyelv használatától, mert rontja a stílust, nehezíti az „igazi" programozás mesterségének (művészetének) elsajátítását.

Mivel a BASIC-re sokan felnéznek (például a TV-BASIC-ből vizsgázott több ezer ember), és számosan lenézik azt, időszerűnek látszik a nyelv tényleges helyének meghatározása, érdemeinek és hátrányainak felmérése, egybevetése. A téma aktualitására utal a hazai BASIC irodalom jelentős mértékű szaporodása, valamint az a tény is, hogy idén áprilisban az NJSZT megfelelő szakosztályának műsorán is szerepelt.

A hagyományoknak megfelelően először a BASIC előnyeit vizsgáljuk, és megpróbáljuk elejét venni egyes — elsősorban a korszerűségét kétségbe vonó — kritikáknak.

A BASIC kialakulásának dátuma alapján első generációs nyelvnek számít. Születésének idejét 1964-re teszik, a szülők (J. G. Kemény és T. E. Kurtz) egyike magyar. Ekkor világszerte az első és második hardvergenerációs gépek működtek, a tipikus tárméret néhány ezer bájt — bár ekkor még főleg szószerkezetű operatív tárakat használtak —, az átlagos műveleti sebesség pár száztól pár ezer művelet/sec-ig terjedt. A nyelvek közül a BASIC elődje—kortársa például a FORTRAN, a COBOL és az ALGOL—60.

Elfogadva, hogy a nyelv az elmúlt 20 évben korszerűsödött, új utasításokkal bővült, szabad-e a modernségét számon kérni, a jóval később „felfedezett" strukturált programozás támogatásának hiányát felróni egy olyan nyelvnek, amely túlélte az ALGOL-t, pedig ennek bölcsője körül ott bábáskodott a világ akkori számítástudományi elitje, P. Naurtól J. W. Backuson keresztül. A. van Wijngaardenig? Nyilván nem.

A BASIC legfontosabb erénye az egyszerűség. Ez a tulajdonság kétarcú: könnyű a nyelvet akár előzetes ismeretek hiányában is elfogadható szinten elsajátítani (a TV-BASIC tanfolyam adásideje 15 × ½ óra) és viszonylag nem nehéz egy-egy konkrét gépen megvalósítani sem.

Az átlag felhasználót az elsajátíthatóság érinti közvetlenül, számára az egyszerű megvalósíthatóság abban nyilvánul meg, hogy már 4 kilobájtos, zsebméretű gépeken is használhatja a BASIC-et.

A nyelvet azért könnyű megtanulni, mert kevés, egyszerű fogalommal dolgozik, utasításkészlete szerény, az utasítások pedig közérthető, tipikus tevékenységeket hajtanak végre.

Az egyszerű megvalósíthatóság nagyrészt a nyelv szerkezetén alapszik, amely ahol csak lehet, segíti a gépi feldolgozást. Ennek érdekében írunk egy sorszámot a program utasításai elé (a modernebb változatokban két sorszám között már több utasítás is állhat), és ezért kell a numerikus és szöveges változók neveit egymástól megkülönböztetni (X és X$). Az általában nem nagyon hatékony értelmező rendszer munkáját segíti, hogy minden utasítás egy jellemző kulcsszóval kezdődik, és így azonnal felismerhető. Az értékadásban szereplő LET kulcsszó elhagyását újabban sok értelmező eltűri.

A BASIC további előnye az átlagosnál jobb felhasználói kapcsolat. A rendszer általában egy szerkesztő alrendszert is tartalmaz, amely a már előbb említett sorszámokat jól használja fel a BASIC program sorainak rendezésére. Sok szerkesztő az igen barátságos, ún. „full-screen", azaz a képernyő tetszőleges pontját értékelő módon működik. A BASIC tervezői tehát jól látták, hogy a programozók munkájuk során túlnyomórészt írják vagy módosítják programjukat, ezért ebben kell őket hatékonyan támogatni. A program írása inkrementálisan, utasításról utasításra haladva történik, és minden utasítás elfogadásáról vagy hibájáról a rendszer azonnal felvilágosítást ad. Kár, hogy ettől az igen hasznos megoldástól több népszerű BASIC eltért, például a C64 is.

Ha furcsán hangzik is, úgy tűnik, hogy a BASIC-nek nemcsak erényei, hanem bűnei is az egyszerűségből fakadnak. A nyelv legszembetűnőbb hiányossága, amely a modern programozási elvek érvényesülését is akadályozza, az igazi eljárások hiánya. A GOSUB még a névvel rendelkező, paraméter nélküli eljárásokat sem igen pótolja, pedig egy strukturált programterv lépésenkénti finomítással történő gépi megvalósítása jól használható paraméterátadási mechanizmust igényel. A paraméterek globális változók segítségével való átadása felesleges, a feladat lényegéhez nem tartozó tevékenységeket erőltet a programozóra. Ebből már következik, hogy az IF . . . THEN utasításban igen gyakran szerepel a GOTO, ami a programot áttekinthetetlenné teszi.

Másik jogos vád a BASIC ellen az, hogy az értelmezési technika, ami igen hasznos a rendszer barátságos viselkedése szempontjából, meglehetősen lassú programfutáshoz vezet. Ezzel kapcsolatban nem szabad figyelmen kívül hagyni a BASIC fordítókat és azt a tényt, hogy nagyobb teljesítményű gépeknél a kis hatékonyságú értelmezés sem jelent problémát.

A BASIC egyszerűségéből, valamint a programozó szabadságának biztosításából eredően sok rendszerben a nyelv nem egyértelmű. Általában például nem kötelező szóközt hagyni egy utasítás alkotórészei között. Ha azonban ezt a FOR I=S TO P utasítással próbáljuk megtenni, több ismert rendszer (például a C64) csődöt mond, mivel az S TO P egybeírva a STOP utasítást adja.

Az egyszerű utasítások kifejező ereje többféleképpen is növelhető. Az egyik mód például az általánosított VALUE függvény alkalmazása, amely nemcsak számokat, hanem egy szöveges változóban tárolt tetszőleges aritmetikai kifejezést is elfogad és kiértékel.

Ha mérleget akarunk vonni a BASIC előnyei és hátrányai között, azt is figyelembe kell venni, hogy a nyelv milyen nagy mértékben járult és járul hozzá a számítástechnika terjesztéséhez. Az általam kihozott egyenleg, amely természetesen nem teljesen objektív, okvetlenül pozitív. A BASIC a maga szintjén megfelelő eszköz. Olyan, mint a Trabant a magyar átlagpolgár számára.

Végül — a hasonlatra alapítva — egy jó tanács és egy észrevétel:

Trabanttal nagy hiba lenne Forma—1 futamon indulni.

A jelenlegi trabantosok jó része szívesen átülne egy jobb kocsiba.

BAKOS TAMÁS

Vigyázat! Tolvaj!

Egyre többen dolgoznak Commodore 64 számítógépen, és ezzel egy időben jelentősen gyarapszik a jogtalan másolással terjedő programok száma. Ezen szeretnénk segíteni. Hosszabb-rövidebb ideig meggátolhatjuk a programunk terjedését, ha hatásosan védjük a különböző másolóprogramok ellen.

Mielőtt rátérnénk az e célt szolgáló különféle techikákra, meg kell ismerkednünk a VC 1541-es meghajtó hardver és szoftver felépítésével. Igaz, erről már több könyv is megjelent a Data Becker kiadó gondozásában — például a „Das grosse Floppy—Buch" és a „Floppy VC 1541 Pflegen und Reparieren" —, de ezekből a kiadványokból pontosan a védelemre mutató részek hiányoznak. Most ezeket a hézagokat pótoljuk.

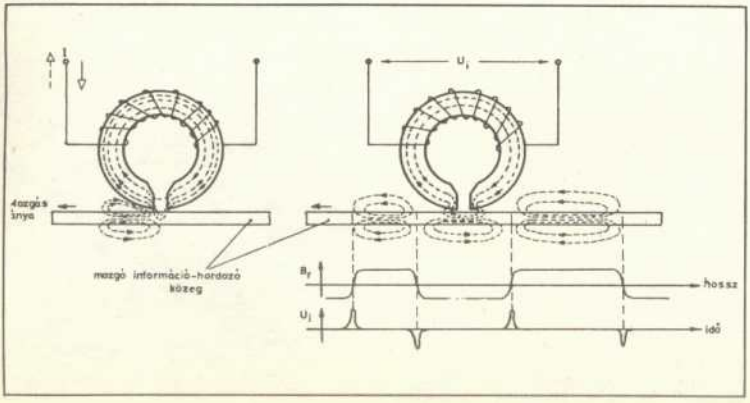
Nézzük először is, hogyan rögzíti a meghajtó a floppylemezen az adatokat. A lemez mágnesezhető bevonata általában vasoxid vagy Ni-Co ötvözetű fémréteg. Jelrögzítéskor ennek a bevonatnak picinyke felületdarabja bemágneseződik, és kis lokális mágnesként viselkedik; nagyságát mikrométerekben mérhetjük.

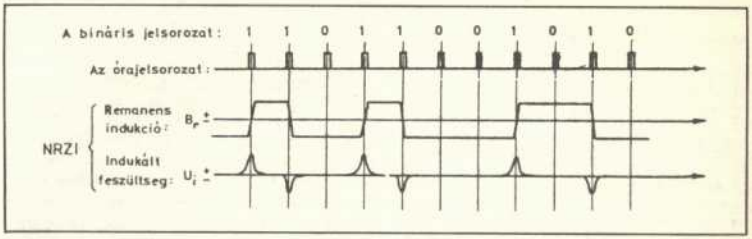
Az író-olvasó fejek sematikus szerkezetét az *1. ábra* mutatja.

Beírásnál áramot vezetünk a tekercsbe. Mivel az információt hordozó közegnek a fejtől mért távolsága kisebb a mag légréséénél, a magban létrejövő mágneses erővonalak a légrés helyett többségében az információhordozó közegen keresztül záródnak. Ezáltal a közegnek azt a részét, amely az íróáram idején a fej alatt mozog, az áramiránynak megfelelő mágneses irányban telítésbe viszi. Miután ez a tartomány elhagyja a fej terét, megfelelő irányú indukció marad benne vissza. Ha az íróáram irányát megváltoztatjuk, az indukció iránya is megváltozik. Az azonos irányú indukcióval rendelkező összefüggő tartományok egy-egy kis mágnest alkotnak.

Kiolvasásnál ezek a kis mágnesek rendre a fej alá kerülnek, és erővonalaik egy része a fej magján keresztül záródik. Ha az erővonalak iránya hirtelen megváltozik, a tekercsben a változás irányának megfelelő polaritású feszültségimpulzus indukálódik.

1. ábra





2. ábra

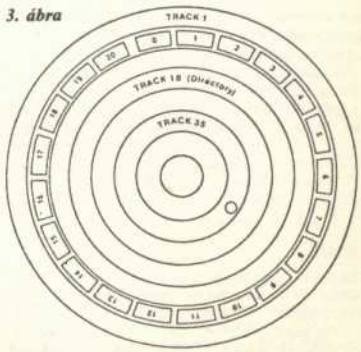
A digitális jelrögzítésre az egyik leggyakoribb módszert, a zérusra vissza nem térő invertáló (Not Return to Zero Inverting; NRZI) eljárást alkalmazták, amelynek jellegzetes képét láthatjuk a *2. ábrán* egy adott bináris jelsorozatra.

Nézzük, ezekből a mágneses jelsorozatokból hogyan is épül fel a mágneslemez adatállománya.

A lemezegység motorja 300 fordulat/perc sebességgel forgatja a lemezt, ami a centrifugális erő hatására elveszti hajlékony jellegét. Így lehetővé válik az író-olvasó és a törlőfej mozgatása egy léptetőmotor segítségével, amely sugárirányban mozgatja a fejeket. Ezzel a fejmozgatással a jelek koncentrikus körökben (sávokban) rögzítődnek. Két sáv távolsága megközelítően 1/40 inch, egy sáv szélessége pedig 1/80 inch. Ilyen módon a DOS (Disk Operating System) 35 sávot (track) használ, amelyeket kívülről és eggyel kezdődően számoz meg a rendszer. A lemezegység léptetőmotorja kétszer ennyi sávot is képes lenne behatárolni, de ez a lemez kapacitását jelentősen megnövelné, és ezt az író-olvasó fej nagysága nem teszi lehetővé. Ugyanis ha a léptetőmotor által megengedett 70 különböző fejállást felhasználnánk — vagyis két sáv közé pozicionálnánk a fejet —, akkor nagy valószínűséggel valamelyik sávon felülírnánk az ott lévő adatokat, tehát az megsemmisülne.

A rendszer a sávokat szektorokra osztja, amelyekbe egy adatblokknyi (256 adatbájt) információt ír (lásd a *3. ábrát*). Fontos, hogy 256 adatbájtról van szó, ugyanis a DOS az adatokat a később bemutatott GCR (Group Code Recording) kódformátumban rögziti, amelyben egy adatblokk 325 bájtnyi.

3. ábra



Mivel a meghajtó — a legtöbb típustól eltérően — a külső sávokba sűrűbben ír, ott több szektor fér el; ilyen módon négy zónára osztja a lemezt, amelynek adatait az *1. táblázat* mutatja.

1. táblázat

Zóna	Sávszám	Szektorszám	Szektorok száma	Bájt
1	1—17	0—20	21	5376
2	18—24	0—18	19	4864
3	25—30	0—17	18	4608
4	31—35	0—16	17	4352

Látható, hogy standard formattálás esetén 683 blokk található a lemezen.

A meghajtó a szinkronjel segítségével írja fel az adatokat a lemezre. A különböző írássűrűséget úgy éri el, hogy a 16 MHz-es órafrekvencia 13-as, 14-es, 15-ös és 16-os osztásával állítja elő az előbb említett szinkronjelet, így a négy zónában más-más időközönként kapja az impulzust. A *2. táblázat* a pontos adatokat mutatja.

2. táblázat

Sáv	Az órajel leosztása	Két impulzus között eltelt idő
1—17	16 MHz/13 = = 1,230769 MHz	0,8125 µs
18—24	16 MHz/14 = = 1,142857 MHz	0,8750 µs
25—30	16 MHz/15 = = 1,066666 MHz	0,9375 µs
31—34	16 MHz/16 = 1 MHz	1 µs

Ennek megfelelően egy bit, illetve egy bájt felírási sebessége a *3. táblázat* szerint alakul.

3. táblázat

Sáv	1 bit felírási ideje	1 bájt felírási ideje	Bit/s	Bit/fordulat
1—17	3,25 µs	26 µs	307 692	61 538,4
18—24	3,50 µs	28 µs	285 714	57 142,8
25—30	3,75 µs	30 µs	266 667	53 333,4
31—34	4,00 µs	32 µs	250 000	50 000,0

A bitsűrűség 4000 bit/inch és 6000 bit/inch között mozog. Ha az egész lemezen a megengedett legnagyobb írássűrűséggel dolgoznánk, akkor a teljes lemez kapacitása 2 027 676 bit, vagyis 253 459 bájt lenne. A váltakozó írássűrűség miatt a lemez tényleges kapacitása csak 174 848 bájt, amiből a felhasználó 169 984-et (256 × 664) tölthet fel, ugyanis a többit a rendszer foglalja le a lemezkatalógus számára.

Hogyan szabályozhatjuk az írássűrűséget? Erre a célra a VIA 6522 chip B kapujának 5. és 6. bitje (PB5, PB6) hivatott, amelyet lemezre íráskor vagy olvasáskor a Disk controller állít be folyamatosan.

Az egyes bitkombinációk jelentését a *4. táblázat* tartalmazza.

4. táblázat

Sáv	1 bit írássebessége	PB5	PB6
1—17	3,25 µs	0	0
18—24	3,50 µs	0	1
25—30	3,75 µs	1	0
31—35	4,00 µs	1	1

Ezeknek az adatoknak a birtokában már két védelmi elvet is megérthetünk. Mindkét módszerben közös, hogy a lemez egy részét a standard eljárástól eltérően formattáljuk. Mivel a lemezformázás módja sokféle lehet, csekély a valószínűsége, hogy a másolóprogram éppen az általunk használt módszer szerint dolgozik.

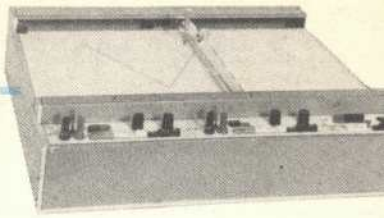
Elsőként készíthetünk egy olyan programot, amellyel megváltoztatjuk a lemez egyes sávjain az írássűrűséget, és így biztosítjuk, hogy a DOS ne olvashassa el. Ahhoz viszont, hogy mi hozzáférjünk, nem elég az íróprogramot elkészíteni, hanem egy olvasóprogramot is kell írni, amellyel az adatokat majd visszaolvassuk. Hogy kinek melyik sáv szimpatikus, melyiken kívánja a formátumot megváltoztatni, mindenki maga dönti el; ezért ha a védés módszerét ismerjük is, mégsem lehet a lemezt lemásolni, mert ehhez tudni kellene, hogy melyik sáv formátumát változtatták meg, továbbá azt is, hogy ezen a sávon az adatok milyen sűrűn helyezkednek el.

A második eljárás is a lemez némely sávjának formátumát módosítja, de az adatsűrűség változatlan marad. Említettük, hogy bár a léptetőmotor több mint 70 sáv pozicionálására is képes, az író-olvasó fej mérete nem teszi lehetővé ennek kihasználását, mert ha két sáv közé írunk, akkor a határsávokon megsérülhetnek az adatok. Ezt a két sávot viszont nyugodtan feláldozhatjuk a titkosítás érdekében: így olyan területet nyerünk, amelyet csak a saját dekódoló programunk olvashat, tehát más nem férhet hozzá.

Ahhoz, hogy a fejet beállíthassuk akár két sáv közé is, vagy hogy nagyobb bitsűrűséggel tudjunk írni a lemezre, még sok mindennel meg kell ismerkednünk. Ezt sorozatunk további részei remélhetőleg elősegítik.

KOVÁCS P. ATTILA

Gyakran kell számítógépekkel olyan feladatokat megoldanunk, amelyeknek eredményeire valamilyen rajzok elkészítéséhez van szükség. Kényelmes, ha a gép által kiszámított adatokból nem nekünk kell a pontokat, vonalakat rajzeszközökkel, például görbe vonalzókkal megrajzolni, megszerkeszteni — ez az út egyébként is fáradságos, lassú, nem mindig kielégítően pontos, és sok a hibalehetőség is —, hanem a rajzolást is géppel végeztethetjük el. Ilyenkor a számítógép kiszámítja a rajzoláshoz szükséges adatokat, és a rajzológép működését is vezérli.



Az EMG 79812 típusú rajzológépe

A gépi rajzolás, a regisztráló berendezések alkalmazása jóval korábbi technikai vívmány a modern számítástechnika rajzológépeinél; igaz, az előbbiek túlnyomórészt analóg jellé alakítható mennyiségeket regisztráltak.

A fejlődés ellenére a rajzológép (nem rajzgép!) — angol nevén plotter — nem volt mindig olyan gyakran használt számítógép-perifériális egység, mint napjainkban. A rajzológép mai tömeges elterjedtségéhez a vizuális megjelenítési igény fokozódása mellett a rajzológépek árának jelentős csökkenése is hozzájárult. Ma már olyankor is készíttetünk rajzokat, ha ez nem kötelező, pusztán azért, hogy numerikus eredményeinket szemléletes formában tanulmányozhassuk, hiszen az emberi agy — tudósok szerint! — 60 000-szer gyorsabban fogja fel a grafikusan megjelenített információt, mint ugyanazt számok, illetve szöveg formájában.

Ezt könnyen és gyorsan megtehetjük, mert már tömegével állnak rendelkezésünkre olcsó és elfogadható pontosságú, számítógéphez, sőt még otthoni számítógépekhez is csatlakoztatható rajzológépek.

Igaz, hogy rajzolásra majdnem minden nyomtató alkalmas valamilyen mértékig. Számos rajzolási feladatot nem is érdemes igazi rajzológépekkel elvégeztetni, hiszen (nem precíziós célokra!) megfelelő rajzokat kaphatunk egy közepes „pontosságú" és „felbontóképességű" tűs, lézer vagy akár mágneses vagy tintacseppes nyomtató alkalmazása révén is.

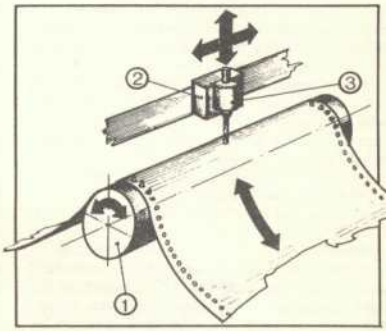
Vannak azonban olyan pontosságra, vonalminőségre és rajzméretre vonatkozó követelmények, amelyeket csak rajzológépekkel tudunk kielégíteni. Foglalkozzunk tehát röviden (a teljesség igénye nélkül) a rajzológépekkel kapcsolatos tudnivalókkal!

Jelenleg két rajzológép-konstrukció tömeges gyártása folyik. Az egyikben a rajzfelület (papír, pausz, film vagy fémlemez) rögzítve van a rajzolás kezdetétől annak befejezéséig. Ilyen például az EMG 79812 típusú vagy a RADELKIS OH—860 rajzológépe.

A másik konstrukcióban a rajzfelület is mozgást végez a géphez képest.

A rögzített rajzfelületű rajzológépeknél téglalap alakú sík felületre (asztalra) rögzítik a rajzhordozót. A rögzítés rászívással vagy elektrosztatikus úton szokott történni. A rászívásnál az asztal meghatározott helyein apró (kb. 0,1 mm átmérőjű) furatok vannak, amelyeken keresztül állandóan szívják a levegőt. Az így létrejött vákuum rögzíti a papírt vagy más rajzhordozót.

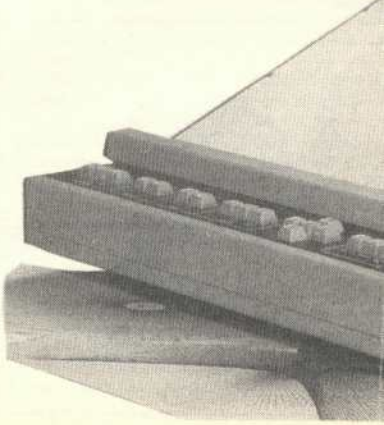
Az elektrosztatikus rögzítés a különböző előjelű elektromossággal feltöltött testek közötti vonzóerőt használja fel a papír rögzítésére. Az előbbiekben említett RADELKIS OH—860 és az EMG 79812 típusú gépek ezt a rajzhordozó-rögzítési módot használják.



1. ábra. Az IMI EPA—207 rajzológép konstrukciója

Ahhoz, hogy rajzológépünkkel rajzolhassunk, szükséges egy rajzolószerkezet is, amely a rajzeszközt befogó részből, valamint magából a rajzeszköz(ök)ből áll.

A RADELKIS OH—860 típusú rajzológépe



Rajzológépek

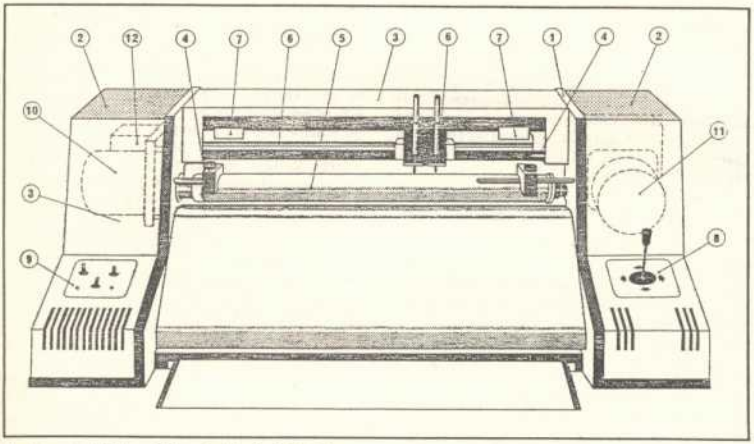
A több rajzeszközt tartalmazó rajzolószerkezet befogónyílásaiban a rajzeszközök egyenes vagy körív mentén helyezkednek el.

A rajzeszközökkel szemben komoly követelmény az, hogy nagy sebességű rajzolásnál is folyamatos és egyenletes vastagságú vonalat húzzanak.

A különböző papír alapanyagú rajzfelületekre általában rosttollal, golyóstollal, igényesebb esetekben speciális csőtollal rajzolnak. A belső borítónkon látható zebra rajza golyóstollal készült. Elképzelhető az is, hogy nemcsak rajzeszközöket fognak be a rajzolószerkezetbe, hanem különböző szerszámokat. Például üveglapok vagy fémlapok gravírozásához karcoló tűket, műanyagok esetén különböző fejtőkéseket használnak.

A rajzolószerkezet mozgatására különböző módszereket ismerünk.

A rögzített rajzfelületű gépeknél a téglalap alakú asztal egyik átellenes oldalpárján kialakított sínekre támaszkodik két végével egy mozgatható rúd (ún. „híd"), amely mindig csak az asztal másik oldalpárjával párhuzamos helyzetben lehet. E rúdon is van egy sín, amelyen e rúd mentén mozoghat az íróeszköz tartó kocsi. Az elmondottakat jobban megérthetjük, ha az esztergapadok hossz- és keresztszánjaira vagy a híddaruk hídjára és futómacskájára gondolunk.



2. ábra. Az EPA—207 rajzológép felépítése

Vannak azonban más, például acélhúros mozgatással működő rajzológépek is. Ezeknél a típusoknál a hossz-szán, illetve a kocsimozgatást végző keresztszán sínjei helyett acélhúrokat alkalmaznak. Az ilyen megoldású rajzológépek pontossága romlik az acélhúrok bizonyos idő után történő megnyúlása következtében. Az acélhúros mozgatás pontatlanságát küszöböli ki a menetes orsóval vagy fogasléccel történő szánmozgatás. Ezek természetesen precíziós megmunkálást, valamint a hőtágulás okozta méretváltozásokat kiküszöbölő mechanikai rendszert igényelnek.

A rajzolószerkezet és a papírmozgatás céljaira általában léptetőmotorokat használnak. A léptetőmotorok olyan tulajdonságúak, hogy az állórészükre adott vezérlőimpulzus hatására forgórészük mindig ugyanakkora „egységnyi" szögelfordulást végez. Így a forgórész mozgása szakaszos, nem akárhol állhat meg. A léptetőmotor forgásiránya is megváltoztatható. Az egy impulzus hatására bekövetkező szögelfordulás, az ún. lépésszög értéke általában 1° és 30° közé esik.

Ahhoz, hogy megfelelő pontossággal mozgathassuk a rajzolószerkezetet, bonyolult elektronikával kell vezérelnünk a léptetőmotorokat, és precízen megmunkált mechanikára van szükség.

A rögzített rajzfelületű gépek rajzméretei kb. A4-es mérettől szobányi területűig terjedhetnek. A berajzolni kívánt felület határvonalait általában programtechnikai úton jelölhetjük ki oly módon, hogy megadjuk a bal alsó, illetve a jobb felső pontok koordinátáit. Erre rendszerint külön utasítások szolgálnak.

A rajzológépek másik nagy csoportját az ún. hengeres vagy dobos rajzológép — angol nevén drum plotter — képezi. Ez a típus kisebb pontossági követelmények kielégítésére szolgál. Elnevezése onnan származik, hogy a rajzhordozóul alkalmazott papírfelület egy henger alakú dobról tekeredik le. Ilyen elven működik például az EPA—207 típusú rajzológép, amelyet az Ipari Műszergyár Iklad (IMI) gyárt. A belső borítónkon szereplő ábra is ezzel a dobos rajzológéppel készült. Ezt a gépet a gyár vendégeként volt alkalmunk közelebbről is tanulmányozni.

A konstrukció alapelvét az *1. ábra* szemlélteti. A papír síkján belüli hosszirányú mozgatást az ①jelű papírmozgató dob végzi a papírtovábbító tüskék segítségével. Az oldalirányú (dob alkotó menti) tollmozgatást a ②jelű írófej végzi. A tollaknak a papírhoz való érintését és felemelését a ③jelű tollmeghúzó elektromágnes végzi.

A működési alapelv az, hogy a léptetőmotorok megfelelő fogaskerék-áttételen keresztük csatlakoznak a papírmozgató dobhoz, illetve a tollakat tartó írófejhez. Ha csak a dobot mozgatjuk — és a toll a papírral érintkezik — „függőleges" egyenest kapunk. Fordított esetben — ha csak az írófejet mozgatjuk — rá merőleges, „vízszintes" egyenest rajzol a toll. „Ferde" egyenes rajzolása mindkét motor működtetésével lehetséges, amikor is az egyenesek szöge a léptetőmotorok sebességarányától függ. Például 45°-os egyenes rajzolásakor mindkét motor egyforma segességgel forog.

A *2. ábrán* láthatjuk a rajzológép felépítését. Az ①a váz, a ②a műanyagból készült oldalborító lemez, a ③ fedőlemez védi az írófejet és az azt vezető sínt, az ugyancsak ③jelű papírterelő lemez a papír „megvezetéséről" gondoskodik. A ④ papírleszorító karok a papírtovábbító tüskékre szorítják a papírt. ⑤a papírmozgató dob. A ⑥rajzolófejbe két tollat foghatunk be.

A ⑦végálláskapcsolók érzékelik a rajzolófejek szélső helyzetét, így behatárolhatják az oldalirányú rajzméretet. A ⑧ botkormánnyal (kézikapcsolóval) a gép off-line üzemmódjában módunkban áll a toll helyzeteit előzetesen beállítani, vagy segítségével akár „kézzel" is rajzolhatunk nyolc főirányban.

A ⑨jelű rész a vezérlő előlap, ahol beállíthatjuk az üzemmódokat. A kezelő visszajelzéseket is kap a beépített világítódiódák (LED-ek) révén.

A dobmozgató egység a ⑩ jelű rész, ellenpárja a ⑪jelű léptetőmotoros rajzolófejmozgató egység. A rajzolófej mozgatása ennél a típusnál bordás szíjmeghajtással történik. ⑫a szíjfeszítő ház.

A rajzológép hátoldali részén — ez ábránkon nem látszik — helyezkedik el a hálózati betáplálás a biztosítóházzal és a kapcsolóval, valamint a gép vezérléséhez szükséges illesztőcsatlakozó. Az illesztőcsatlakozó kétirányú CCITT V24-es és RS232 csatlakozást tesz lehetővé, max. 128 bájt hosszúságú utasítások, illetve adatblokkok részére. Ezen keresztül illeszthető például PROPER 8—16, IBM PC, HT—1024 és Commodore gépekhez is.

SZEBENSZKI SÁNDOR

„Vigyázó szemetek Párizsra vessétek!"

SICOB — volt és lesz!

A párizsi SICOB — mely 1949. évi alapítása óta a francia és az európai informatika elsőrangú bemutatójává fejlődött — 1986 őszén már-már szegényesedni látszott. A kiállítók száma 850-ről 600-ra csökkent, és a hagyományosan 10 napos kiállítás tavaly szeptemberben csak egy hétig tartott nyitva. A jelenség oka azonban nem más volt, mint visszahúzódás a nagy „ugrás" — terjeszkedés — előtt.

A SICOB kinőtte a CNIT (Centre National des Industries et des Techniques) Defense negyedbeli területét, és — saját hirdetésének szavait idézve — „elröpül" Villepinte-be. Villepinteben, Párizstól északra ugyanis 1986 októberében, az 5-ös számú csarnok átadásával befejeződött Európa legmodernebb kiállítási központjának, a Parc International D'Expositions Paris-Nordnak a felépítése. Ezzel lehetővé vált, hogy 1987 tavaszán a SICOB 117 100 m²-en fogadja látogatóit.

A megnövekedett kiállítási terület, az új környezet, valamint az egyidejűleg megrendezendő és tematikájukkal azt kiegészítő események kitűnő alkalmat kínálnak a SICOB-nak: bővítve a bemutatott termékek körét, a világ egyik legnagyobb vásárává és üzleti központjává léphet elő.

1987. április 6—11. között Párizsban az új Nemzetközi Vásárközpont és a Porte Maillot-nál levő Kongresszusi Palota hat esemény színhelye lesz. Ezek a következők:

1. Maga a hagyományos SICOB

Nemzetközi kiállítás, melyen több mint 120 ország kiállítói mutatják be termékeiket négy ipari főcsoportban: irodaügyviteli eszközök; számítástechnika (hardver és szoftver, mikro is!); távközlési eszközök; irodaszervezési eszközök.

2. Irodaszervezési konferencia

1987-ben először hívják össze Franciaországban. Célja, hogy új metodikákat, technológiákat mutasson be, melyekkel kényelmesebbé tehetik környezetüket az ügyviteli rendszerek felhasználói.

3. Alkalmazástechnikai konferencia

Előadásai bővebb ismertetést nyújtanak a SICOB-on kiállított termékekről, azok gyártási technológiájáról, a gyártmányok főbb felhasználási területeiről.

4. Informatikai konferencia

Már szintén hagyományos: Európa legnagyobb szoftverkonferenciája.

5. Automatizálási konferencia

Az 1987. évi SICOB alkalmával először rendezi meg a GIIPRA (a francia vállalatok szövetsége). Az automatizálási szakemberekhez kíván szólni arról, hogyan élhetnek a számítógép adta lehetőségekkel speciális problémáik megoldásában.

6. SSII 87 — SICOB szolgáltatás és mérnöki informatika

Ez a szintén az idén bemutatkozó kiállítás főleg a szolgáltató és tanácsadó vállalatok munkájához nyújt segítséget azzal, hogy közepes és nagy rendszerekhez ajánl alap- és felhasználói szoftvert, szoftvercsomagokat.

A SICOB átköltözése Villepinte-be az eddigi gyakorlat megfordítását jelenti. Eddig ugyanis volt egy ún. tavaszi SICOB májusban, amit speciális SICOB-nak neveztek, és volt egy „nagy" SICOB ősszel. 1987-ben pontosan fordítva lesz. Az 1987. áprilisi után, 1987. szeptember 15—20-ig a CNIT-ben lesz megtartva a SICOB Special Micro, egy időben a Kongresszusi Palotában rendezett Infodial-Videotex kiállítással. Így az idei év különösen gazdag programot kínál az informatikai szakembereknek. A SICOB-bal közel egy időben ugyanis Lyonban az INFORA, Hannoverben pedig a CEBIT is csábító lehetőség a tájékozódásra.

A 37. SICOB, mely utolsó volt a maga nemében, az alábbi fő területeken mutatott újdonságokat:

— a szupermikrók megjelenése,
— az AT kompatibilis rendszerek fejlődése,
— helyi (lokális) hálózatok fejlesztése,
— grafikai munkahelyek (munkaállomások) tökéletesedése,
— nyomtatók.

A *szupermikrók* — külső megjelenésükre olyan „dobozok", melyekben már nemcsak a VC kártyákat és az ellenőrző egységeket építették be, hanem a Winchester-lemezeket és a védőegységet is — egy vagy több multiprocesszort tartalmaznak, gyorsak (innen származik a nevük), ellátják a Unix operációs rendszer vagy annak valamely változata vezérlését, és képesek a felhasználók tucatjait több funkcióban kiszolgálni. Ilyen kategóriájú gépeket a kisebb francia cégek közül a Sistral, az Interntechnique, a Mercure és a Plessey France ajánlott az őszi SICOB-on. Annak ellenére, hogy az USA-ban ezek a típusok már a legnagyobb sikert aratták, itt csak kisebb érdeklődést váltottak ki. Az idei SICOB fogja eldönteni, hogy meg tudják-e vetni „lábukat" az európai piacon is.

Ugyancsak nagy választékkal jelentek meg a mikro családhoz tartozó PC, XT, AT kompatibilis, hordozható Mac típusok. Ezek között az ADDX-System kínálta azt, amely a piacon az egyik legjelentősebb terméknek mutatkozik; az AT kompatibilis ADD—X 286 rendszer 3. modellje 80286-os, 8-10 MHz-es mikroprocesszorra épül, kiegészíthető 1—9 Mbájtos RAM-mal, a lemez kapacitása 20—71 Mbájt, a védőegységé 20 Mbájt.

A *hordozható típusok* nagy száma egyébként jellemző volt a mikrók más kategóriáiban is. E téren az úttörő szerepet a Tandy, a HP és a Data General játszotta. A Data General az 1986-os SICOB-on mutatta be a DG/One új változatát, a 2. számú modellt, amely tetszés szerint működtethető folyadékkristályos vagy normál képernyővel.

Másik érdekes vetülete volt az 1986-os kiállításnak a nagyszámú *grafikus munkaállomás* megjelenése, amelyek jól tükrözik a mikroprocesszorok fejlődését. Legtöbbjük 68020-as mikroprocesszorra épül, nemcsak a kielégítő sebességi paraméterek miatt, hanem főleg azért, mert ezek speciális áramkörei alkalmazhatók a legjobban a grafikában és a CAD-ban. Az élenjáró kiállítók a Gixi, a Sun Microsystems és a Control Data voltak ebben a kategóriában. Például a Gixi cég a Radiance 3020/3030 jelű munkaállomását mutatta be. Ez 68000-es 10 MHz-es áramkörrel és 68881-es kalkuláló processzorral, 3 D-s gyorsítóval, 19 hüvelykes 1024 × 800 pontos színes képernyővel, 1—4 Mbájtos memóriával van felszerelve. Ára 160 000 FF körül volt az alapváltozatban.

A *távközlésben* szenzációnak számított az x25 és x32 kártya, a Transpac vevő, az ATLAS 400-as üzenetrögzítő megjelenése, valamint a helyi hálózatok nagy száma. A *helyi hálózatok* fejlesztésének imponáló példáját mutatta be az ADDX-System. Az ADD X-Link egy helyi hálózat buszra csatlakoztatva, Novell logikai egységgel. Az RLX Link lehetővé teszi nyolc RLX rendszer egymás közötti kapcsolatát.

A *technológiai* érdekességekről szólva, nagy figyelmet érdemeltek az optikai információhordozók (optikai lemez, videolemez, CD—ROM) között a CD—ROM első kereskedelmi prototípusai, illetőleg azok első ipari alkalmazásai a Philips és Sony cégeknél. A holnap fő memóriájának, a CD—ROM-nak további gyártói között 1987-ben várhatóan már francia cég is jelentkezik.

Az 1986-ban kiállított *nyomtatók* közül két modell újszerűsége keltett érdeklődést. Az AST France, amely a világon először állított elő személyi számítógéphez is használható kártyákat és perifériákat, most a Turbo lézerrel jelent meg a piacon. Ez az összes PC-vel kompatibilis, olyan asztali lézernyomtató, amely gazdaságos komplett egységként működtetve, és újság minőségben képes nyomtatni, továbbá határozott vonalú, egész oldalas grafikákat rajzol. Memóriája 1,5 Mbájt, amely a kontroll kártyára könnyedén diszponálható. Saját kategóriájában az AST Turbo lézer különösen gyors, percenként 8 oldalt nyomtat. Ezt a teljesítőképességét az AST a lézeres vezérlésnek (LPC) köszönheti. A Turbo lézer az összes PC logikával kompatibilis, beleértve a táblázatkészítőket, a szövegszerkesztőket, az alapadatokkal manipuláló kereskedelmi grafikákat éppúgy, mint a CAO és DAO grafikákat. Ára 12 000 FF, egyéves ingyenes karbantartással.

Egyik konkurens a Siemens, amely egy egészen más — a PT—85-ös hőátadásos — elven működő nyomtatójával jelentkezett. Nyomtatvány minőségű terméket ad, biztosítja a szöveg magas szintű variálhatóságát, és grafikák megörökítésére is képes. A hőátadásos technológia mellett a nyomtatás tetszőleges papírra — a normál papírtól az átlátszó fóliáig — kerülhet. Könnyedén cserélhető a betűk formája is. A PT—85 kompatibilis az IBM PC-vel, ára 9170 FF volt. Jóval olcsóbb a lézernyomtatónál, viszont nem olyan gyors.

Az 1986-os ősz legjellegzetesebb vonása mégsem a fent említett új termékek, technológiák megjelenése, hanem az IBM kompatibilis gépek árainak zuhanása volt, ami 40 százalékra tehető a múlt év folyamán. A SICOB reagált a válsizre, és a Villepinte-be való átköltözéssel együtt felkészült arra, hogy meghatározó szerepet töltsön be a világ számítógépiparára jellemző főbb tendenciák bemutatásában.

SÁGHINÉ dr. HARTAI VERONIKA

A DIGITÁL rejtvénypályázat eredményhirdetése

Kedves Fiatalok!

Rejtvénypályázatunk lezárultával közöljük az egyes fordulók feladatainak megoldásait.

Mielőtt ezt megtennénk, engedjétek meg, hogy néhány mondatban összefoglaljuk -jó tanács jelleggel- tapasztalatainkat.

A beküldött megoldásokból leszűrhető, hogy nagy lelkesedéssel láttatok hozzá a programíráshoz. Talán ebből a tényből adódtak a jellemző hibák is. Sokan nem ismeritek -vagy csak nem alkalmazzátok- a programciklusokat.

Többen nem vettétek figyelembe a feladatkiírást /futtathatóság C16-on és HT-n a második forduló feladatainál, az adatok sztringekben történő megadása a harmadik forduló feladatainál/.

Aki programot ír, annak nem szabad megfeledkezni néhány fontos szempontról sem:

- a program legyen jól tervezett,
- hatékony
- áttekinthető /más számára is érthető
- működőképes

Az értékelés módja:

A feladatokat az első kivételével azonos súlyúnak tekintettük, függetlenül a nehézségi fokban mutatkozó kisebb különbségektől. A megoldásokat pontoztuk. Az 1. feladatra 5, a többire - hibátlan megoldás esetén - 10 pontot adtunk.

A különleges vagy "elegáns" megoldásokat 1-2-3 további ponttal jutalmaztuk. Értékelési szempontnak tekintettük azt is, hogy mind a három forduló feladatait beküldtétek-e.

Az értékelés eredményeként a nyertes:

TURCSÁN ZSOLT /2941 Ács, Kertész u. 3./ lett

34 ponttal, aki a DIGITÁL Számítástechnikai Szaküzletben /1o26 Bp. Szilágyi Erzsébet fasor 35./ választhatja ki a kifrásban felsoroltak közül a neki legmegfelelőbb nyereményt. A nyertesnek gratulálunk!

Dicséret illeti Nyéki Pétert /Kemenesszentpéter, Jókai u.48./, aki 28 pontot, valamint Adamát Lajost /Kecskemét, Jókai u.1./, aki 26 pontot ért el.

További munkátokhoz sok sikert kíván a Szerkesztőség !

A feladatok helyes megoldása:

Első forduló 1. feladat:

A program a fokban megadott szög szinuszát számítja ki a sin x hatványsorával történő közelítéssel, 10^{-12} pontossággal.

```
10 FORI=0TO25:PRINT:NEXT:REM ERNYOTORLES
20 PRINT"CIMLETEZES":RESTORE
30 INPUT"AZ OSSZEG";P:IFP<0THEN30
40 FORI=1TO9
50 READA:K=INT(P/A):P=P-K*A
60 IFK<>0THENPRINTK;"DARAB";A;"FORINTOS"
70 IFP<>0THENNEXTI
80 INPUT"UJRA (I/N)";A$
90 IFA$="I"THEN10
100 IFA$="N"THEN120
110 GOTO80
120 END
130 DATA1000,500,100,50,20,10,5,2,1
```

2. feladat: A címletező program

```
10 REM ATVITEL A USER PORTON
20 REM AZ ADO PROGRAMJA
30 OUT31,7:OUT30,255
40 PRINT"A KARAKTER?"
50 X$=INKEY$:IFX$=""THEN50ELSEX=ASC(X$)
60 X=ASC(X$)
70 OUT31,14:OUT30,1
80 OUT31,15:OUT30,X
90 OUT31,14:OUT30,0
100 OUT31,15:OUT30,0
110 GOTO40
```

Második forduló: Az adó programja

```
10 REM ATVITEL A USER PORTON
20 REM A VEVO PROGRAMJA
30 OUT31,7:OUT30,0
40 OUT31,14:A=INP(31)
50 IFA<>1THEN40
60 OUT31,15:B=INP(31)
70 PRINT"AZ ADAT:";CHR$(B)
80 GOTO40
```

A vevő programja

Harmadik forduló: A digitális órát szimuláló program

```
10 DEFSTRA:J=0:DATA31,28,31,30,31,30,31,31,30,31,30,31
20 RESTORE:INPUT"HO,NAP";A:IFLEN(A)<>4THEN20
30 HN=VAL(LEFT$(A,2)):NP=VAL(RIGHT$(A,2))
40 IFHN<1ORHN>12THEN20
50 FORI=1TOHN:READB:NEXTI:IFB<NPTHEN20
60 INPUT"ORA,PERC,MASODPERC";A:IFLEN(A)<>6THEN60
70 H=VAL(LEFT$(A,2)):M=VAL(MID$(A,3,2)):S=VAL(RIGHT$(A,2))
80 IFH<0ORH>23ORM<0ORM>59ORS<0ORS>59THEN60
90 IFINKEY$<>""THENJ=NOTJ:A=""
100 IFJ=-1THENCLS:PRINTUSING"###";HN;NP:GOTO120
110 CLS:PRINTUSING"###";H;M;S
120 FORI=0TO380:NEXT
130 S=S+1:IFS=60THENS=0ELSE90
140 M=M+1:IFM=60THENM=0ELSE90
150 H=H+1:IFH=24THENH=0ELSE90
160 NP=NP+1:IFNP>BTHENNP=1:READBELSE90
170 HN=HN+1:IFHN>12THENHN=1:RESTORE
180 GOTO90
```

ADOK – VESZEK – CSERÉLEK

Ebben a rovatban rövid, szöveges, a mikroszámítógépekkel kapcsolatos hirdetéseket közlünk. A díjszabás: közleteknek gépelt soronként (60 karakter) 100,— Ft, magánszemélyeknek az első sor 50,— Ft, minden további sor 20,— Ft. Az NJSZT tagjainak az első három sor ingyenes. Hirdetéseiket a szerkesztőség címére várjuk.

■ 48 k-s Spectrumhoz játékprogramokat cserélek. A válaszokat listával kérem. Cseh Attila, Debrecen, Kapitány u. 16. 4033.

■ C64 programdokumentációkat keresek, cserélek. Cserébe 1985/86. évi fejlesztésű szuper játékprogramokat, grafikai programokat (dokumentációval), zene- és sakkprogramokat adok. Programcsere is lehetséges — kizárólag lemezen. Honti József, Csákvár, Május 1. út 11. 8083.

■ Vállalatok, figyelem! Programozást, rendszertervezést, számítástechnikai szaktanácsadást vállalok. Egyedi programrendszerek készítése magas színvonalon C64, IBM PC kompatibilis gépekre. Telefon: 154-352.

■ Commodore 16-os játékprogramokat cserélnék. Sutus Péter, Bp. Kelli u. 9. fszt. 1. 1135.

■ C64 programokat cserélek kazettán. Laczkó Attila, Sülysáp, Dózsa György út 1/c 2241.

■ Commodore 64 magnó, joystick, könyvekkel és programokkal eladó. Buzinszki Attila, Miskolc, Vasverő út 4. 3535.

■ Commodore VIC—20 magnóval, assembler, disassembler, monitor és játékprogramokkal, német, magyar nyelvű dokumentációval 10 000 forintért eladó. Oláh István, Szolnok, Kassai u. 56. I/4. Telefon: 56-15-496.

■ EPROM-programozó készülékhez 28 lábas IC-foglalatok eladók (390,— Ft/db). Csányi Gábor, Mezőtúr, Ifjúsági lkt., 13. III. 16.

■ C64-es programokat cserélek. Cím: ifj. Sitkei Zoltán, 7200 Dombóvár, Gunaras kemping. Tel.: 74/65-523.

■ Floppy diszk (kétoldalas, dupla sűrűségű) eladó. Tel.: 425-242.

■ ZX81 6 k-s bővítővel, szakkönyvekkel, programokkal 8000 forintért eladó. Tel.: 896-227. Cím: Krajcsovits László, 1043 Budapest, Virág u. 23. II. 16.

■ Commodore 64-es játékprogramokat cserélek. Bárdos Péter, Borsodbóta, Rákóczi út 162. 3658.

A haldoklónak hitt Sinclair QL többek csodálkozására feléledt és táncra perdült.

Armstrad, aki szőröstül-bőröstül felvásárolta a Sinclair céget, 400 nyugatnémet márka körüli áron kiárusítást tart. Ez az alacsony ár természetesen sok új rajongót szerez a gépnek. A lelkes tábor és a néhány mecénás, akik a gép továbbfejlesztett változatának gyártására is vállalkoztak, elősegítik a QL reneszánszát.

Nálunk is felkérték táncra a QL-t: a tavalyi 300-zal szemben ugyanis napjainkban már 600 QL van hazánkban, és megalakult a QLub-juk is. A Csokonai Művelődési Ház (Bp. XV., Eötvös u. 64-66.) adott otthont a QLub-nak, amely még csak 33 fős létszámmal, de igazi lelkesedéssel indult. Itt a csatlakozni szándékozóknak is módjukban áll, hogy társakkal találkozzanak, egyelőre minden szerdán délután 5 és 9 óra között. *(Faragó—Pacher)*

Ki ad magyarázatot?

A C16 egy közepes hosszúságú, kb. 8 kbájtos BASIC program újrasorszámozásakor különös dolgot produkált. A RENUMBER parancsra szintaktikai hibát jelzett, és a program fele „deformálódott".

A programot csak úgy tudtam helyreállítani, hogy Monitor üzemmódban átírtam a veremmutatót (SP regiszter) $ F8-ra, így ismét kilistázhattam a programot, de a GOTO-k paraméterei átszámozódtak, és csak hosszas munkával kaptam vissza.

IFJ. MONFERA RÓBERT

Helyreigazítás

Sajnálatos módon az 1986/10. számunkban közölt Programtitkosítás című cikkbe hiba csúszott. A P—3 helyfoglaló program leírása említi a Ø-ás utasítássor szerepét. A program listájáról azonban ez a sor lemaradt:

Ø GOTO 2

Az utasítás szerepe kettős: egyrészt átlépi az 1-es utasítássort, ahol a program lefutása után már nem szabályos BASIC-kódok lesznek, másrészt a 15Ø-es sor csak így tudja az 1-es sor REM-je utáni területre tölteni a gépi kódú rutint.

A szerző elnézést kér mindazoktól, akik sikertelenül próbálkoztak a programmal. Kéri, ne adják fel; ezzel a kis kiegészítéssel a várt eredmény nem fog elmaradni.

Adom a magyarázatot!

A lap 1986. októberi számában megjelent „Ki ad magyarázatot?" kérdésre válaszolok. A felvetett jelenségek magyarázata a következő.

1. A jelenséget a Primo BASIC interpreterének programhibája: egy állapotjelző helytelen kezelése okozza. Ezt a hibát a Primo—B típusnál már kijavították. Az INPUT, INPUT# és a READ utasítások ugyanis csak a bemenő adatok forrásában (billentyűzet, magnó, DATA sor) különböznek egymástól. Ezért célszerű az utasításokat egy olyan rutinnal kezelni, amely ezt a különbséget figyelembe tudja venni. Ehhez viszont valahogy jelezni kell, hogy melyik utasítás hajtódik végre. Erre két bájt szolgál. Az egyikből kiderül, hogy READ vagy INPUT utasítás zajlik-e: tartalma INPUT esetén zérus, READ utasításnál nem zérus. A bájt a 16606 címen található. A másik bájt (címe 16553) azt jelzi, hogy az INPUT végrehajtásakor a billentyűzetről vagy a magnóról — pontosabban a magnópufferből — kell-e olvasni. Magnós értéke zérus, a billentyűzetnél pedig nem zérus.

A READ utasítás egy DATA mutatót használ, amely megadja, hogy a következő adat melyik címen található. Ettől a címtől kezdve beolvassa a szükséges adatokat, majd a mutató új értékét tárolja a következő beolvasáshoz. Nos, a hiba itt következik be!

Az új érték tárolása ugyanis attól függ, hogy a 16553 cím tartalma 0-e. Ha igen, akkor nem tárolja az új értéket. Ezért tűnik úgy, mintha a READ után RESTORE lenne, mert minden READ utasítás ugyanarról a címről olvassa az adatot, hisz a mutató nem kapja meg az új értéket. Az eddigiekből következik, hogy a hiba nemcsak szöveges változóknál, hanem mindig jelentkezik.

Ezt a 16553 című bájtot a READ utasítás nem állítja be magának zérustól különböző értékre. INPUT# utasításnál viszont értéke 0, és az INPUT# sem változtatja meg az utasítás kezelése után. A hiba tehát nem a felhasználó BASIC programjában van, de szerencsére könnyen kivédhető. Amikor INPUT# után READ-et kérnénk a programban, akkor a READ előtt adjunk ki egy POKE 16553,1: utasítást. Ezután a cikk a—c pontjában írtakra nincs szükség.

2. A problémát az adatok felvétele és beolvasása közötti időzítéskülönbség okozza. A folyamatot példán mutatom be. Van mondjuk egy „A" nevű, 5000 bájtos numerikus tömbünk. Tipusa közömbös. Ha ezt a tömböt a fájl megnyitása után a FOR I=1 TO 5000: PRINT# A(I): NEXT: CLOSE: ciklussal vesszük fel, akkor a következő zajlik. A PRINT# utasítás egy 256 bájtos pufferbe írja az adatokat addig, míg a puffer be nem telik. Ez gyorsan bekövetkezik. Ekkor a puffer tartalmát egy blokkban felveszi a magnóra, és folytatja a most már üres pufferbe a kiírást. Addig ismétlődik a dolog, míg az adatok el nem fogynak. A fájlt lezáró utasítás a még pufferben levő részt is felveszi.

Ezáltal lesz egy valahány blokkból álló felvételünk, ahol a blokkok között alig észrevehető szünetek vannak. Hiba akkor keletkezik, ha a visszaolvasás nem ugyanolyan ciklussal halad, mint a felvétel, mivel a Primo nem tudja vezérelni a magnót. Ha beolvasáskor például a beolvasott adattal valamilyen műveletet is végzünk, akkor egy adat feldolgozása hosszabb ideig tart, mint a felvétele. Ezáltal a program nem üríti ki a puffert arra az időre, mikor a magnóról már a következő blokk adatai jönnek. Így ez a blokk elvész, hiszen a gép nem akarja beolvasni. Nem biztos, hogy az elvesztés rögtön hibaüzenetet okoz. Az adatfájl végénél viszont hiányozni fognak adatok, és ez már FD ERROR jelzést eredményez.

Tapasztalatom szerint még ugyanolyan ciklusú visszaolvasásnál is ajánlatos a biztonság kedvéért egy rövid, üres FOR ciklust tenni a PRINT# utasítás után a kiírásnál. Ha a visszaolvasást más módon végezzük, mint a felvételt, akkor a FOR ciklust olyan hosszúra válasszuk, hogy a felvételen az egyes blokkok között 2-3 másodperces szünetek legyenek. Ezekben a szünetekben kényelmesen megállíthatjuk a magnót.

E módszer szépséghibája, hogy nem tudhatjuk, a magnót mikor kell újraindítani. Én ez úgy oldottam meg, hogy a kapcsolási rajz X4 mezőjében levő T2 tranzisztort a szükséges ellenállásokkal beépítettem a gépbe, és egy LED-et tettem rá. Ez mindig világít, ha a gép olvasni akar.

Enyhén szólva nem szép a gyártótól, hogy az ilyen hibákat nem publikálja!

A „Prima ötlet"-hez is van egy megjegyzésem. Mint a szerzők is írják: „elég kezdetleges", bár több a semminél. A RESET gomb megnyomása azonban több problémát okoz, mint amit a cikkben említenek. Hatására ugyanis a régi képernyőterületen elhelyezkedő adatok — kellően nagy program esetén maga a program vagy az adatai — is kitörlődnek. Ez 6 kbájtnyi információ elvesztését jelenti, ami igen nagy bosszúságot és sok plusz munkát okozhat. Ezért javaslom, hogy aki elvégzi ezt a bővítést, az mindenképpen végezze el a szükséges EPROM-cserét is.

Ezzel és az előbbi LED-beépítéssel kapcsolatban is szívesen segítek mindenkinek.

SZABADOS ISTVÁN
Bp., Szabadság u. 48/a 1028

A Spectrum operációs rendszere ZX81-en

Az alábbiak azoknak az érdeklődésére tarthatnak számot, akik ZX81 számítógépüket nem akarják más típusra kicserélni, de szeretnének gépükön Spectrum BASIC-ben dolgozni, esetleg a Spectrum gazdag programkínálatából egyes gépi kódú programokat is — lehetőség szerint minimális munkaráfordítással — gépükön futtatni.

Mint látni fogjuk, erre van lehetőség. Természetesen a gép nem fog tudni színes képet szolgáltatni, hangja sem lesz, de ettől eltekintve, meglehetős hasonlóságot fog mutatni a Spectrumhoz, beleértve a finomgrafikát is. Mindezt azonban nem kapjuk ingyen: elég sokat kell dolgozni, annál többet, minél nagyobb hasonlóságot akarunk megvalósítani a két gép között. A munkához segédeszközök is kellenek: valamilyen monitor-disassembler a ZX81-re, esetleg assembler is; jó, ha van egy ZX81 ROM-listánk, magyarázatokkal (12); elengedhetetlen a Spectrum ROM-lista és egyéb, a Spectrumra vonatkozó segédkönyvek (7, 8, 9, 10).

Némi készség szükséges a hardvermódosítások elvégzéséhez. A ZX81 kapcsolási vázlatát például (11) adja meg. Az ismertetendő legegyszerűbb változathoz is 64 k-s memóriabővítő kell, melyet ráadásul a finomgrafika megvalósításához át kell alakítanunk. Akinek Memopak 64 k-s bővítője van, az az átalakítást karcolótű, huzal és forrasztópáka segítségével el tudja végezni. Az átalakításhoz (1) szakirodalom ad segítséget. A nálunk nagy számban található, az Audio Computers 32 k-s memóriájáról másolt bővítő is átalakítható 64 k-sra.

Memóriabővítés

A memóriabővítővel szemben két követelmény van. Egyrészt a finomgrafika megvalósításához az RFSH jel alatt a RAM-ból olvasni kell tudnunk; másrészt a 0000H—3FFFH, eredetileg ROM-tartomány helyére a RAM ugyanezen tartományát kell tudni kapcsolni. Ebben a tartományban kell ugyanis majd a Spectrum átalakított, illetve kiegészített ROM programjának elhelyezkednie. Ha ráadásul ezt a részt írásvédetté tesszük a monitorprogram bekerülése után, akkor az esetleges „elszállások" után nem kell mindig megismételni a programbetöltést.

Mivel a két gép közötti alapvető különbség a képalkotásban van, tekintsük át röviden, hogyan történik ez az egyes esetekben. A ZX81-et igen olcsó gépnek szánták, ezért minél több feladatot igyekeztek szoftver úton megoldani, hogy a gép szerkezete minél egyszerűbb lehessen. Így a képalkotásban is jelentős szerepe van a processzornak. Ezt a feladatot azonban csak egyéb feladatai rovására tudja elvégezni; ennek következtében a „hasznos" programok lassabban futnak, mint akkor futnának, ha minden idejét rájuk tudná fordítani. A sebességcsökkenés jelentős. Az ún. SLOW üzemmódban, amikor állandóan van kép, csak kb. negyed sebességgel dolgozik a gép a FAST üzemmódhoz képest, amikor viszont csak akkor van kép, ha a gép feladatát már elvégezte, illetve amikor billentyű megnyomására vár. Ezenfelül egyéb korlátozásokat is tudomásul kell vennünk. Gépi kódú program csak a 0000H—7FFFH tartományban futhat, bizonyos regisztereket (IX, AF') állandóan a képalkotás céljaira kell rendelkezésre bocsátanunk, a maszkolható és nem maszkolható megszakítást is ez használja.

Másként áll a dolog a Spectrumnál. Itt a képet hardver úton állítjuk elő, a fenti korlátozások jórészt elesnek. A monitorprogramnak mindössze arról kell gondoskodnia, hogy a képinformációt tartalmazó RAM-rész — az ún. képernyőfájl — folyamatosan fel legyen töltve a megjeleníteni kívánt új információval. A képalkotás bizonyos feltételek mellett itt is sebességcsökkenést okoz, ez azonban nem túlzottan jelentős.

Feladatunkat most már úgy oldjuk meg, hogy a Spectrum monitorprogramot kiegészítjük olyan képalkotó programmal, mely a ZX81 adottságait felhasználva, a Spectrum képernyőfájl tartalmát meg tudja jeleníteni a képernyőn; a monitorprogram azon részeit, melyek megszakításokat és beviteli-kiviteli műveleteket tartalmaznak, szükség esetén átalakítjuk; végül az így módosított programot a RAM említett helyére juttatjuk, és átadjuk neki a vezérlést.

A „képernyőrutint" (1) alapján lehetett létrehozni. A cikkben majdnem minden lényeges gondolat megtalálható, mely célunk eléréséhez szükséges volt.

A járulékos programokat a Spectrum monitorprogram fel nem használt 386EH—3CFFH területén helyezzük el, itt csupa FFH található. A módosításokat úgy végezzük, hogy lehetőleg olyan cím, amelyet valamely, a Spectrumhoz kidolgozott program hívhat, ne változzék.

A gyakorlati használathoz igen lényeges, hogy a magnetofonról a programokat gyorsan tudjuk betölteni; a ZX81 eredeti SAVE—LOAD eljárása erre a célra alkalmatlan, mivel lassú és megbizhatatlan. A választás a c't folyóiratban közölt Supertape programra esett, mely a ZX81-re (2)-ben található. Ezzel a programmal 3600, illetve 7200 baud sebességgel rögzíthetünk programot szalagra.

Néhány tanács a Supertape program használatához. Célszerű lehet a szalagra való felvétel szintjét növelni. Ez úgy történhet, hogy a ZX81-ben R29-et 1 Mohm-ról 10 kohm-ra változtatjuk, C12-t áthidaljuk, C11-et és R27-et eltávolítjuk. Az „Ear" bemenet érzékenyítése R34 eltávolításával történik (3). Ugyanekkor C10 értékét meg kellett növelni 0,33 µF-ra: ezt kis kapcsolóval célszerű kiiktathatóvá tenni, mert az eredeti ZX81 impulzusalakhoz ez sok lehet. Bár az idézett cikk szerint a Supertape a közönséges magnetofonokon is jó eredményt ad, jobb eredményt lehet elérni, ha magnetofonunkat a digitális üzem követelményeinek megfelelően átalakítjuk (4). Fontos a mechanika, különösen a szalagtovábbító tüske rezgésmentessége; jelentős javulást észleltünk, amikor ennek műanyag perselyét erősen viszkózus, hőálló zsírral kentük. Nagyon jó eredményt ad a Polimer C60-as szalag.

A Supertape a Spectrumon is megvalósítható. Ha tehát mód van rá, készítsünk a Spectrum ROM-ról szalagra Supertape formátumú felvételt, s ez a ZX81-be bejátszható. A másik lehetőség, hogy olyan programot állítunk össze, mellyel az eredeti Spectrum-felvétel a ZX81-be tölthető. A következőkben ezt ismertetjük.

Elöljáróban bemutatok egy segédprogramot *(1. program)*, mellyel a szalagon levő program impulzusainak vagy szüneteinek hosszát tudjuk megállapítani. Jó szolgálatot tesz akkor is, amikor a felvétel-lejátszás minőségére vagyunk kíváncsiak.

A programmal 8000 H—BFFFH-ig folyamatosan feltöltjük a memóriahelyeket a szünet- vagy az impulzushosszra jellemző számmal. Rövid program esetén azt többször betöltjük vagy a betöltési tartomány felső címét csökkentjük. A program indítása: FAST után RAND USR 16514.

Ha a szünetek helyett az impulzushosszakra vagyunk kíváncsiak, a 4088H és 4090H helyen levő utasításokat ellenkező értelműre változtatjuk. Ha katalógusból kikeressük az egyes utasításokhoz tartozó végrehajtási időket, akkor a tényleges szünet-, illetve impulzusidőket is kiszámíthatjuk. A ZX81 órajel-frekvenciája 3,25 MHz.

Egy JVC gyártmányú rádiómagnón egy eredeti, digitális magnón felvett Spectrumprogram a szünethosszakra a következő eredményeket adta:

Bevezető fütty:	3A—3F
Szinkronjel:	10
Program 0 szint	15—18
1 szint	2E—35

Az impulzus-, illetve szünethosszak tehát szórnak. A szórás nagysága az adatrögzítő

```
100 REM    4082  210080  LD HL,8000 ;a betoltes kez
                                    deti cime
110 REM    4085  DBFE    IN A,(FE)  ;lekerdezzuk az
                                    EAR bemenetet
120 REM    4087  17      RLA        ;a 7.bit-a bem.
                                    szint-Carrybe
130 REM    4088  38FB    JR C,4085  ;ha a szint ma-
                                    gas,vissza
140 REM    408A  0E00    LD C,00    ;szunet kezd.:
                                    hossztarolo re
                                    giszter 0
150 REM    408C  0C      INC C      ;hossztarolo re
154 REM                             ;giszter inkrem
160 REM    408D  DBFE    IN A,(FE)  ;EAR bemenet?
170 REM    408F  17      RLA        ;allapot C-be
180 REM    4090  30FA    JR NC,408C ;ha alacsony a
                                    ;szint,vissza
190 REM    4091  79      LD A,C     ;A-ba a hossz-
                                    ;jellemzot
200 REM    4092  77      LD(HL),A   ;(HL)-be
210 REM    4093  23      INC HL     ;kovetkezo ta-
                                    ;rolohely
220 REM    4094  3E8F    LD A,8F    ;8F00-ig tarol-
                                    juk a hosszakat
230 REM    4096  BC      CP H       ;elertuk mar?
240 REM    4097  C8      RET Z      ;ha igen,vege
250 REM    4098  18EA    JR 4085    ;ha nem,vissza
                                    ;az elejere
```

1. program

```
100 REM      1 REM (IDE JON A 3.PROGRAM 182 BAJTJA)
110 REM      2 PRINT "A BETOLTES KEZDETI CIME:"
120 REM      3 INPUT A
130 REM      4 PRINT A
140 REM      5 LET X=INT(A/256)
150 REM      6 LET Y=A-256*X
160 REM      7 POKE 16515,X
170 REM      8 POKE 16514,Y
180 REM      9 POKE 16517,X
190 REM     10 POKE 16516,Y
200 REM     11 LET A=A+18
210 REM     12 LET X=INT(A/256)
220 REM     13 LET Y=A-256*X
230 REM     14 POKE 16519,X
240 REM     15 POKE 16518,Y
250 REM     16 RAND USR 16520
```

2. program

```
100 REM    4088  CD230F  CALL 0F23  ;FAST uzemmod
110 REM    4088  1E00    LD E,0     ;szamlalo reg.
                                    ;nullazasa
120 REM    408D  CD460F  CALL 0F46  ;BREAK-proba
130 REM    4090  D2A603  JP NC,03A6 ;ha BREAK,Dhiba
140 REM    4093  1C      INC E      ;szaml.novelese
150 REM    4094  D3FF    OUT(FF),A  ;magas szint a
                                    ;TV kimeneten
160 REM    4096  7B      LD A,E     ;A-ba a szaml.
170 REM    4097  FEFF    CP FF      ;elertuk mar FF
                                    ;et?
180 REM    4099  300E    JR NC,40A9 ;ha elertuk,va-
                                    ;runk a szink-
                                    ;ronjelre
190 REM    4098  1600    LD D,00    ;a D szaml.=0
200 REM    409D  01804A  LD BC,4A00 ;a futtyszunet
                                    ;<4A
210 REM    40A0  CD0541  CALL 4105  ;a futtyszunet
                                    ;hossza
220 REM    40A3  FE01    CP 01      ;<4A?
230 REM    40A5  20E4    JR NZ,4088 ;ha nem,vissza
240 REM    40A7  18E4    JR 408D    ;ha nem,vissza
250 REM    40A9  D3FF    OUT(FF),A  ;echo a kepen
260 REM    40AB  1600    LD D,00    ;D=0
270 REM    40AD  010022  LD BC,2200 ;a szinkronjel-
                                    ;hossz <22
280 REM    4080  CD0541  CALL 4105  ;a szinkronszu-
                                    ;netet keressuk
290 REM    4083  FE01    CP 01      ;elertuk mar?
300 REM    4085  20F2    JR NZ,40A9 ;ha nem,tovabb
                                    ;keresunk
310 REM    40B7  2A8240  LD HL(4082);adatlerakas
                                    ;itt kezd
320 REM    40BA  1E00    LD E,00    ;paritasellenor
                                    ;zo gyujtoreg.
330 REM    40BC  D3FF    OUT(FF),A  ;echo a kepen
340 REM    40BE  010025  LD BC,2500 ;hosszu es rov.
                                    ;szunet kozt
350 REM    40C1  CD1941  CALL 4119  ;bajtkomplemens
360 REM    40C4  7A      LD A,D     ;A-ba D
370 REM    40C5  2F      CPL        ;a bajt meghat.
380 REM    40C6  77      LD(HL),A   ;a bajt betolt.
390 REM    40C7  AB      XOR E      ;paritaskepzes
400 REM    40C8  5F      LD E,A     ;ezt a gyujtore
                                    ;giszterbe
410 REM    40C9  D5      PUSH DE    ;DE stackre
420 REM    40CA  EB      EX DE,HL   ;DE-be a betol-
                                    ;tes helyet
430 REM    40CB  2A8640  LD HL(4086);HL-be a file
                                    ;vege
440 REM    40CE  A7      AND A      ;Carry=0
450 REM    40CF  ED52    SBC HL,DE  ;HL-DE
460 REM    40D1  EB      EX DE,HL   ;HL-be a betol
                                    ;tes helyet
470 REM    40D2  D1      POP DE     ;DE vissza
480 REM    40D3  2804    JR Z,40D9  ;ha vege,kont
                                    ;roll
490 REM    40D5  23      INC HL     ;ha nincs vege
                                    ;kov.bajt
500 REM    40D6  C3BC40  JP 40BC    ;vissza
510 REM    40D9  78      LD A,E     ;A-ba a parit.
520 REM    40DA  A7      AND A      ;flagallitas
530 REM    40DB  C2A603  JP NZ,03A6 ;ha nem zerus,
                                    ;D hiba
540 REM    40DE  E5      PUSH HL    ;HL stackre
550 REM    40DF  2A8240  LD HL(4082);HL=(4082)
560 REM    40E2  ED5B8440LD DE(4084);DE=(4084)
570 REM    40E6  A7      AND A      ;Carry=0
580 REM    40E7  ED52    SBC HL,DE  ;HL-DE
590 REM    40E9  E1      POP HL     ;HL vissza
600 REM    40EA  2048    JR NZ,4134 ;ha (4082)<>
                                    ;(4084),kesz
602 REM    40EC  23      INC HL     ;kov.bajt helye
604 REM    40ED  228240  LD(4082),HL;most (4082)-be
                                    ;ezt tesszuk
606 REM    40F0  110700  LD DE,0007 ;a filehossz
                                    ;helyenek meg-
                                    ;hatarozasa
608 REM    40F3  A7      AND A      ;flagallitas
610 REM    40F4  ED52    SBC HL,DE  ;HL a low bajt-
                                    ;ra mutat
612 REM    40F6  5E      LD E,(HL)  ;E-be a file-
                                    ;hossz l.bajt
614 REM    40F7  23      INC HL     ;HL a h.bajtra
                                    ;mutat
616 REM    40F8  56      LD D,(HL)  ;D-be a h.bajt
618 REM    40F9  2A8240  LD HL(4082);HL-be a file-
                                    ;kezdet
620 REM    40FC  ED5A    ADC HL,DE  ;CY=0,HL a file
                                    ;vege
622 REM    40FE  23      INC HL     ;a paritasbajt
623 REM    40FF  228640  LD(4086),HL;helye
624 REM    4102  C38B40  JP 408B    ;vissza
626 REM    4105  DBFE    IN A,(FE)  ;EAR=?
628 REM    4107  17      RLA        ;szint Carrybe
630 REM    4108  38FB    JR C,4105  ;ha magas,visz-
                                    ;sza
632 REM    410A  0E00    LD C,00    ;ha alacsony,
                                    ;C=00
634 REM    410C  0C      INC C      ;hosszmero no-
                                    ;velese 1-el
636 REM    410D  DBFE    IN A,(FE)  ;EAR=?
638 REM    410F  17      RLA        ;a szint CY-be
640 REM    4110  D20C41  JP NC,410C ;ha meg ala-
                                    ;csony,vissza
642 REM    4113  79      LD A,C     ;ha valtott,
                                    ;A-ba C-t
644 REM    4114  90      SUB B      ;levonjuk B-t
646 REM    4115  CB12    RL D       ;ha A<B,CY=1,
                                    ;ezt D-be
648 REM    4117  7A      LD A,D     ;A-ba D-t
650 REM    4118  C9      RET        ;visszateres
652 REM    4119  1600    LD D,00    ;a bajt gyujto-
                                    ;regisztere
654 REM    411B  CD0541  CALL 4105  ;1. bit
656 REM    411E  CD0541  CALL 4105  ;2. bit
658 REM    4121  CD0541  CALL 4105  ;3. bit
660 REM    4124  CD0541  CALL 4105  ;4. bit
662 REM    4127  CD0541  CALL 4105  ;5. bit
664 REM    412A  CD0541  CALL 4105  ;6. bit
666 REM    412D  CD0541  CALL 4105  ;7. bit
668 REM    4130  CD0541  CALL 4105  ;8. bit
670 REM    4133  C9      RET        ;vissza
672 REM    4134  CD280F  CALL 0F28  ;SLOW
674 REM    4137  C9      RET        ;visszateres
```

3. program

```
100 REM    4082  CDE702  CALL 02E7  ;FAST
102 REM    4085  3E7F    LD A,7F    ;
104 REM    4087  DBFE    IN A,(FE)  ;BREAK?
106 REM    4089  1F      RRA        ;eredmeny CY-be
108 REM    408A  D3FF    OUT(FF),A  ;echo a kepen
110 REM    408C  38F7    JR C,4085  ;mig meg nem
112 REM                             ;nyomtuk BREAK-
114 REM                             ;et,vissza
116 REM    408E  210080  LD HL,8000 ;forras kezdet
```

és visszaadó lánc jóságának függvénye. Az adatokat statisztikailag értékelve bizonyára érdekes tapasztalatokat lehetne szerezni a magnókra, szalagminőségekre vonatkozóan.

A fenti példát tekintve látható, hogy a fütty-, a szinkron- és a programszintek jól elkülöníthetők. Csakis ilyen esetben remélhetjük, hogy felvételi problémánk nem lesz.

A következő program a Spectrum ROM-program ZX81-be töltésére szolgál. A gépi kódú részt az 1. REM sorban helyeztük el. A BASIC részt a *2. program*, a gépi kódú részt a *3. program* tartalmazza.

Például a Spectrum ROM-program átvételénél, ha a 0000H címnek a 8000H címet akarjuk megfeleltetni, a következőképpen járunk el:

RUN /Newline
32748 /Newline

Megindítjuk a magnót. A képernyőn vékony fekete vízszintes vonalak láthatók. A beolvasás 0/16 üzenettel áll le. 32748-tól a Spectrum-program fejlécét láthatjuk, a tényleges ROM-tartalom 8000H—BFFFH-ig tart.

A Spectrum ROM módosítása

Most már gépünkben van a Spectrum ROM-tartalma, megkezdhetjük annak módosítgatását. Előbb azonban azt a programot ismertetjük, amelynek segítségével a 8000H—BFFFH között elhelyezkedő, módosított Spectrum ROM-programot a 0000H—3FFFH közötti ROM-terület helyett beiktatott RAM-területre másolhatjuk.

A következőkben — feltéve, hogy a Spectrum-programunkon már minden szükséges változtatást végrehajtottunk —, a Spectrum rendszerre való áttérés az említett program segítségével a következő lépésekben megy végbe:

— Betöltjük a gyorstöltőprogramot ZX81 módszerrel.

— Betöltjük az átmásolóprogramot, most és a továbbiakban már gyorstöltéssel dolgozva.

— Betöltjük a módosított Spectrum ROM programot.

— A RAND USR 16514/Newline-nal megindítjuk az átmásolóprogramot: ez először csak egy zárt programhurokban kering, a képernyőre vízszintes mintát rajzol. A programszámláló tartalma mindig a 4000—7FFF közötti részen van, tehát a 0000H—3FFFH szegmenset átkapcsolhatjuk: ROM helyett RAM.

— A BREAK gomb megnyomásával megtörténik az átmásolás (rövid megnyomás legyen!).

— Ismét zárt hurokba kerülünk, a képernyőn az előbbi minta; ekkor azt az átkapcsolást lehet végrehajtani, mellyel lehetővé tesszük, hogy az egész RAM-tartományban futhassanak gépi programok (ld. később).

— Újabb BREAK: a jól ismert © 1982 Sinclair Research Ltd. üzenet jelenik meg, gépünkön a Spectrum BASIC-ben dolgozhatunk.

Az egész áttérési művelet mindenestül kb. két és fél percet vesz igénybe. Az átmásoló programot *(4. program)* egy REM sorban helyezhetjük el.

Most már rátérhetünk a Spectrum ROM program módosítására. Természetesen a képalkotással kezdjük. Két lehetőség között választhatunk: vagy a ZX81 SLOW vagy a FAST mintájára építhetjük föl a képalkotást. Az előbbit választottuk, annak ellenére, hogy így a gép lassabb, és az NMI-t is igénybe vesszük. Viszont a kép nem ugrál a billentyűk megnyomásakor. A programfutást az NMI generátor kikapcsolásával gyorsíthatjuk. Ilyenkor természetesen nincs kép [lásd (5)-öt].

A képalkotó program megértéséhez ismerni kell bizonyos mértékig a ZX81 lelkivilágát. Részletes leírása (6)-ban található. A legfontosabb tudnivalókat röviden áttekintjük.

Mint ismeretes, a televíziós kép sorokból áll, és az új sort szinkronimpulzus indítja. Ezeket a ZX81-ben frekvenciaosztó állítja elő az órajelből: minden 207 órajel után ad ki egy impulzust. Ez kicsit nagyobb frekvenciát ad ki, mint a szabványos 15625 Hz, mely 208-as osztó esetén állna elő. A sorszinkronimpulzusokat egy OUT(FE),A utasítással rá lehet kapcsolni a processzor NMI bemenetére és OUT(FD),A utasítással onnan lekapcsolni. Lekapcsolt NMI esetén egy IN A,(FE) utasítás a tv-kimeneten 0 szintet állít elő, az OUT(FF),A pedig 1-et. Ezek segítségével állítjuk elő a képszinkronjelet.

A videokimeneten levő képinformáció a hasznos kép alatti és feletti üres sorokból, a hasznos kép soraiból, képenként képszinkron-, soronként sorszinkronjelből áll. A hasznos sornak is van baloldalt és jobboldalt egy üres része.

Miközben a számítógép valamely programot dolgoz fel, minden egyes NMI beérkezésekor megvizsgálja — a ZX81 ROM 0066H-nál kezdődő rutinja segítségével —, hogy elérkezett-e már a képkiadás vagy a képszinkron ideje. Erre a célja az AF' regisztert, mint számlálót használja fel. Ha az idő elérkezett, végrehajtja vagy a képkiadást, vagy a képszinkronadást, mely utóbbi alatt a tasztatúrát is lekérdezi.

A képkiadás a részleteket nem tekintve, a következőképpen zajlik le. Végrehajtunk egy ugrást egy, a képernyőfájl első helyénél 8000-rel nagyobb memóriahelyre, vagyis az A15 értéke 1. A nem teljes memóriadekódolás miatt azonban ez a képernyőfájl első helyére történik. Két dologra emlékeztetünk: a képernyőfájl az ábrázolandó karakterek kódját tartalmazza, másrészt „ugrás" utasításlehívást jelent az ugrási helyről, és ekkor az MI jel aktív. 64 k-s bővítőknél a nem teljes címdekódolás így — a C000H—FFFFH tartományban — csak MI aktív állapotában kell hogy megvalósoljon. Így is készülnek.

Ott tartottunk, hogy a képernyőfájl első memóriahelyének tartalma megjelenik az adatbuszon, miközben A15 értéke 1. Ez utóbbi hatására az ULA a következőt teszi. Ha az adatbusz D6 bitje 0, akkor a processzornak NOP utasítást továbbít. Minden „ábrázolható" karakter kódjának D6 bitje 0. Így a processzor NOP utasítást hajt végre, az ULA pedig megjegyzi a képernyőfájlból kiolvasott karakterkódot. Amikor a NOP-on belül a frissítési ciklushoz érünk, a processzor a címsínre az I és R regiszter tartalmát teszi rá. Az I regiszter tartalma a karaktergenerátor táblázat kezdeti címének magas bájtja: 1 EH.

```
118 REM      4091  110000   LD DE,0000 ;cel kezdet
120 REM      4094  010040   LD BC,4000 ;blokkhossz
122 REM      4097  ED80     LDIR       ;masolas
124 REM      4099  3E7F     LD A,7F    ;
126 REM      409B  DBFE     IN A,(FE)  ;BREAK?
128 REM      409D  1F       RRA        ;az eredmeny
130 REM                                ;CY-be
132 REM      409E  D3FF     OUT(FF),A  ;echo a kepen
134 REM      40A0  38F7     JR C,4099  ;ha nincs BREAK
136 REM                                ;vissza
138 REM      40A2  C30000   JP 0000    ;atadjuk a ve-
140 REM                                ;zerlest a SPEK
142 REM                                ;TRUM monitor-
144 REM                                ;programnak
```

4. program

```
118 REM      xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
120 REM      003B  C5       PUSH BC    ;a fo regiszte-
122 REM      003C  D5       PUSH DE    ;stackra
124 REM      003D  E5       PUSH HL    ;
126 REM      003E  2A805C   LD HL(5C8B);HL-be a kep-,
128 REM                                ;vagy kepszink-
130 REM                                ;ronrutin kez-
132 REM                                ;docime
134 REM      0041  76       HALT       ;megszakitas-
136 REM                                ;ra varunk
138 REM      0042  D3FD     OUT(FD),A  ;az NMI genera-
140 REM                                ;tor lekapcs.
142 REM      0044  E9       JP(HL)     ;ugras a kezdo
144 REM                                ;cimre
```

5. program

```
100 REM      0066  F5       PUSH AF    ;AF stackra
102 REM      0067  FD3447   INC(IY+47) ;a szamlalot
104 REM                                ;noveljuk
106 REM      006A  FA6F00   JP M,006F  ;tovabb szamol
108 REM      006D  28CC     JR Z,0038  ;kep vagy kep-
110 REM                                ;szinkron kia-
112 REM                                ;das kovetkezik
114 REM      006F  F1       POP AF     ;stackrol AF
116 REM      0070  C9       RET        ;vissza
```

```
102 REM      3880  00000000000000000   ;32 db NOP uta-
104 REM      3888  00000000000000000   ;sitas,a pszeu-
106 REM      3890  00000000000000000   ;do videosor
108 REM      3898  00000000000000000   ;
110 REM      38A0  C30339   JP 3903    ;vissza a prog-
112 REM                                ;ramba
114 REM      xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
116 REM      38C0  00000000000000000   ;32 db NOP uta-
```

```
118 REM      38C8  0000000000000000  ;sitas,a 2.psze
120 REM      38D0  0000000000000000  ;udosor
122 REM      38D8  0000000000000000  ;
124 REM      38E0  C31539   JP 3915  ;vissza a prog-
126 REM                             ;ramba
128 REM      38E3  00       NOP     ;
130 REM      38E4  ED73805BLD(5C80),SP;atmenetileg ta
132 REM                             ;roljuk SP-t
134 REM      38E8  31843A   LD SP(3AB4);az IR reg.tab-
136 REM                             ;lazatara mutat
138 REM      38EB  0603     LD B,03  ;
140 REM      38ED  10FE     DJNZ 38ED ;idokitolto
142 REM      38EF  010707   LD BC,0707 ;B=07,C=07
144 REM      38F2  210018   LD HL,1800 ;H-ba a karak-
146 REM                             ;tersorok szama
148 REM      38F5  00       NOP     ;idokitolto
150 REM      38F6  D1       POP DE  ;IR aktualis
152 REM                             ;teke
154 REM      38F7  00       NOP     ;idokitolto
156 REM      38F8  7A       LD A,D  ;A-ba D
158 REM      38F9  ED47     LD I,A  ;I-be az aktu-
160 REM                             ;alis ertek
162 REM      38F8  3E00     LD A,00  ;idokitolto
164 REM      38FD  7B       LD A,E  ;A-ba E
166 REM      38FE  ED4F     LD R,A  ;R akt.ert.
168 REM      3900  C38B88   JP 8838 ;a pontsor kia-
170 REM                             ;dasa (allitott
172 REM                             ;A15)
174 REM      3903  14       INC D   ;a kov.pontsor
176 REM      3904  7A       LD A,D  ;A-ba D
178 REM      3905  05       DEC B   ;a 7.sort kiad-
180 REM                             ;tuk mar?
182 REM      3906  C2F738   JP NZ,38F7 ;ha nem,vissza
184 REM      3909  41       LD B,C  ;B kezdoerteke
186 REM                             ;=07
188 REM      390A  25       DEC H   ;a sorszamlalo
190 REM                             ;csokkentese
192 REM      390B  ED47     LD I,A  ;I-be A
194 REM      390D  3E00     LD A,00  ;idokitolto
196 REM      390F  7B       LD A,E  ;A-ba E
198 REM      3910  ED4F     LD R,A  ;R-be az akt.
200 REM                             ;ertek
202 REM      3912  C3C888   JP 88C0 ;a 8.sor kidasa
204 REM                             ;allitott A15
206 REM      3915  20DF     JR NZ,38F5 ;ha meg nem ad-
208 REM                             ;tuk ki a kepet
210 REM                             ;vissza
212 REM      3917  ED78805BLD SP(5C80);SP eredeti er-
214 REM                             ;teke
216 REM      3918  FD3647C9LD(IY+47)C9;az ures sorok
218 REM                             ;szamaloja
220 REM      391F  213339   LD HL,3933 ;az uj vissza-
222 REM                             ;teresi cim
224 REM      3922  22805C   LD(5C80),HL;ezt (5C80)-ba
226 REM      3925  000000            ;idokitolto
228 REM      3928  DD23     INC IX   ;idokitoltok
230 REM      392A  DD2B     DEC IX   ;
232 REM      392C  D3FE     OUT(FE),A ;bekapcsoljuk
234 REM                             ;az NMI gen.-t
236 REM      392E  E1       POP HL   ;a fo regiszte-
238 REM      392F  D1       POP DE   ;rek vissza
240 REM      3930  C1       POP BC   ;
242 REM      3931  F1       POP AF   ;
244 REM      3932  C9       RET      ;a kep kesz,
246 REM                             ;vissza
248 REM      3933  2A785C   LD HL(5C78);kepvaltasok
250 REM                             ;szama
252 REM      3936  23       INC HL   ;noveljuk
254 REM      3937  22785C   LD(5C78),HL;uj ertek
```

```
256 REM      393A  7C       LD A,H   ;elertuk a 0000
258 REM      393B  B5       OR L     ;erteket?
260 REM      393C  2006     JR NZ,3944 ;ha nem,ugrunk
262 REM      393E  FD3440   INC(5C7A) ;FRAMES-3 nov.
264 REM      3941  C34839   JP 3948  ;ugras
266 REM      3944  00000000000000    ;7 db NOP(kesl)
268 REM      3948  0000              ;kesleltetes
270 REM      394D  E5       PUSH HL  ;
272 REM      394E  E1       POP HL   ;
274 REM      394F  23       INC HL   ;
276 REM      3950  00       NOP      ;kesleltetes
278 REM      3951  DBFE     IN A,(FE) ;kepszinkron
280 REM                             ;kezd
282 REM      3953  00       NOP      ;idokitolto
284 REM      3954  0660     LD B,60  ;
286 REM      3956  10FE     DJNZ 3956 ;kesleltetes
288 REM      3958  D3FF     OUT(FF),A ;kepszinkron
290 REM                             ;vege
292 REM      395A  0603     LD B,03  ;
294 REM      395C  10FE     DJNZ 395C ;kesleltetes
296 REM      395E  FD3647C9LD(IY+47)C9;az ures sorok
298 REM                             ;szamaloja
300 REM      3962  21E438   LD HL,38E4 ;az uj vissza-
302 REM                             ;teresi cim
304 REM      3965  22805C   LD(5C80),HL;taroljuk
306 REM      3968  D3FE     OUT(FE),A ;bekapcsoljuk
308 REM                             ;az NMI-t
310 REM      396A  CD8F02   CALL 028F ;a tasztatura
312 REM                             ;lekerdezese
314 REM      396D  E1       POP HL   ;
316 REM      396E  D1       POP DE   ;a fo regiszte-
318 REM      396F  C1       POP BC   ;rek vissza
320 REM      3970  F1       POP AF   ;
322 REM      3971  C9       RET      ;vissza a fo-
324 REM                             ;programba
326 REM      3AB4  7F401F403F405F40  ;az IR reg.
328 REM      3ABC  FF409F408F40DF40  ;kezdo erte-
330 REM      3AC4  7F481F483F485F48  ;kelnek tab-
332 REM      3ACC  FF489F488F48DF48  ;lazata
334 REM      3AD4  7F501F503F505F50
336 REM      3ADC  FF509F508F50DF50
```

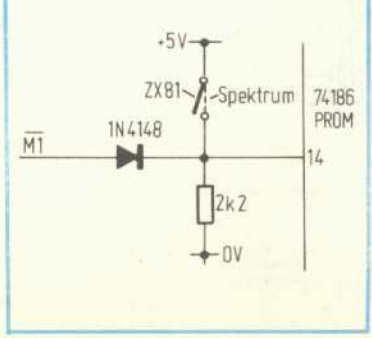
6. program

7. program

```
100 REM      0000  D3FD     OUT(FD),A ;inicializalas
102 REM      0002  AF       XOR A     ;B87F folott
104 REM      0003  117F88   LD DE,887F ;nem toroljuk
106 REM      0006  1808     JR 0013   ;a tarat,mert
108 REM                             ;a pszeudosorok
110 REM                             ;ott vannak
112 REM      xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
114 REM      0013  C3C811   JP 11C8
116 REM      xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
118 REM      0038  C9       RET       ;az INT-ot nem
120 REM                             ;hasznaljuk
122 REM      xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
124 REM      03E1  0000     NOP,NOP   ;Beeper-ben
126 REM      xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
128 REM      03F6  00       NOP       ;a magnetofonke
130 REM      04C6  21C00F   LD HL,0FC0 ;zelo rutinok
132 REM      04CD  214C06   LD HL,064C ;modositasa
134 REM      xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
136 REM      04D4  76       HALT
138 REM      04D5  D3FD     OUT(FD),A
140 REM      04D7  00       NOP
142 REM      04D8  10FE     DJNZ 04D8
144 REM      04DA  D3FF     OUT(FF),A
```

Közben az ULA a megjegyzett karakterkódból és egy hárombites számláló tartalmából kiindulva a ROM címbusz alsó 9 vonalát módosítja a processzor által képezett címhez képest, úgy, hogy az az ábrázolandó karakter megfelelő sorának „bitmintáját" tartalmazza. A ROM számára az ULA által kiadott cím a döntő; az alsó 9 címvonal a processzorétól ellenállásokkal el van választva. Vagyis az ULA a „bitmintát" kapja meg az adatvonalakon, beteszi egy shiftregiszterbe, és a következő NOP alatt a képernyőre kiadja. És ez így megy tovább, míg a képernyőfájlban egy olyan utasítást nem találunk, melynek D6 bitje 1 (például a HALT, melynek kódja 76H). Az ilyen utasításokat az ULA eredeti alakban engedi a processzorhoz. A HALT után az R regiszter nemsokára nullázódik; legalábbis a

1. ábra



6-os bitje, melynek adatvonala a processzor $\overline{INT}$ bemenetével van összekötve. A bekövetkező maszkolható megszakítás a következő sor kiadását készíti elő. Az $\overline{INT}$-ot Sinclair bizonyára csak azért használta fel, mert az 1 k-s gépben a képernyőfájl „komprimálva" van. Mi az $\overline{INT}$-ot nem fogjuk használni, ezért ennek részleteibe nem is megyünk bele.

Az előzőekből a mi képalkotó programunk megértéséhez a következőket szűrjük le:

— az $\overline{NMI}$ jelet OUT(FD),A utasítás kapcsolja le a processzorról,

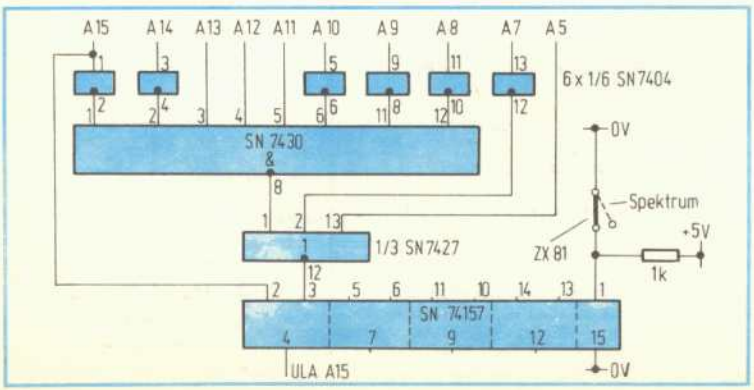
— az $\overline{NMI}$ jelet OUT(FE),A utasítás kapcsolja rá a processzorra,

— lekapcsolt $\overline{NMI}$ esetén az IN A,(FE) a videokimenetet Ø-ra, OUT(FF),A pedig 1-re állítja,

```
146 REM        04DC  00        NOP
148 REM        04DD  0699      LD B,99
150 REM        04DF  10FE      DJNZ 04DF
152 REM        04E1  3E7F      LD A,7F
154 REM        04E3  DBFE      IN A,(FE)
156 REM        04E5  C3DB39    JP 39DB
158 REM        04E8  062A      LD B,2A
160 REM        04EA  10FE      DJNZ 04EA
162 REM        04EC  D3FF      OUT(FF),A
164 REM        04EE  3E7F      LD A,7F
166 REM        04F0  0633      LD B,33
168 REM        04F2  10FC      DJNZ 04F2
170 REM        04F4  DBFE      IN A,(FE)
172 REM        04F6  C37239    JP 3972
174 REM        04F9  0000000000
176 REM        ××××××××××××××××××××××××××××××××××××××××
178 REM        0508  0000
180 REM        ××××××××××××××××××××××××××××××××××××××××
182 REM        050F  C30505    JP 0505
184 REM        0512  0000
186 REM        ××××××××××××××××××××××××××××××××××××××××
188 REM        0518  063D      LD B,3D
190 REM        ××××××××××××××××××××××××××××××××××××××××
192 REM        051C  D3FF      OUT(FF),A
194 REM        051E  0639      LD B,39
196 REM        0520  00        NOP
198 REM        0521  C8        RET Z
200 REM        0522  C3F239    JP 39F2
202 REM        ××××××××××××××××××××××××××××××××××××××××
204 REM        052D  062E      LD B,2E
206 REM        052F  062E      LD B,2E
208 REM        0531  7A        LD A,D
210 REM        0532  3C        INC A
212 REM        0533  C2FE04    JR NZ,04FE
214 REM        0536  0636      LD B,36
216 REM        0538  10FE      DJNZ 0538
218 REM        053A  C9        RET
220 REM        053B  00000000
222 REM        053F  F5        PUSH AF
224 REM        0540  3E7F      LD A,7F
226 REM        0542  DBFE      IN A,(FE)
228 REM        0544  1F        RRA
230 REM        0545  D3FF      OUT(FF),A
232 REM        0547  D3FE      OUT(FE),A
234 REM        0549  00000000000000
236 REM        ××××××××××××××××××××××××××××××××××××××××
238 REM        0558  0000
240 REM        055A  15        DEC D
242 REM        055B  76        HALT
244 REM        055C  D3FD      OUT(FD),A
246 REM        ××××××××××××××××××××××××××××××××××××××××
248 REM        0565  E640      AND 40
250 REM        0567  0000
252 REM        ××××××××××××××××××××××××××××××××××××××××
254 REM        059F  00000000
256 REM        ××××××××××××××××××××××××××××××××××××××××
258 REM        05F6  E640      AND 40
260 REM        ××××××××××××××××××××××××××××××××××××××××
262 REM        05FD  00000000
264 REM        0601  D3FF      OUT(FF),A
266 REM        ××××××××××××××××××××××××××××××××××××××××
268 REM        091E  00
270 REM        ××××××××××××××××××××××××××××××××××××××××
272 REM        098F  CD0A3A    CALL 3A0A
274 REM        0992  0000
276 REM        ××××××××××××××××××××××××××××××××××××××××
278 REM        1187  CDBF39    CALL 39BF   ;a NEW-ban
280 REM        ××××××××××××××××××××××××××××××××××××××××
```

```
282 REM        11CC  00000000000
284 REM        11D2  F3        DI
286 REM        11D3  ED56      IM 1
288 REM        11D5  3E3F      LD A,3F
290 REM        ××××××××××××××××××××××××××××××××××××××××
292 REM        122E  FD213A5CLD IY,(5C3A)
294 REM        1232  CD6E38    CALL 386E
296 REM        ××××××××××××××××××××××××××××××××××××××××
298 REM        18D2  3604      LD (HL),04  ;villogo karak
300 REM                                    ;ter helyett
302 REM                                    ;inverz
304 REM        ××××××××××××××××××××××××××××××××××××××××
306 REM        2298  0000                  ;BORDER-ben
308 REM        ××××××××××××××××××××××××××××××××××××××××
310 REM        33FB  117738    LD DE,3877  ;a konstansok
312 REM                                    ;atlepese ru-
314 REM                                    ;tinban
316 REM        ××××××××××××××××××××××××××××××××××××××××
318 REM        386E  21E438    LD HL,38E4
320 REM        3871  22B05C    LD(5CB0),HL
322 REM        3874  D3FE      OUT(FE),A
324 REM        3876  C9        RET
326 REM        ××××××××××××××××××××××××××××××××××××××××
328 REM        3972  08        EX AF,AF'
330 REM        3973  060A      LD B,0A
332 REM        3975  10FE      DJNZ 3975
334 REM        3977  062C      LD B,2C
336 REM        3979  6F        LD L,A
338 REM        397A  00        NOP
340 REM        397B  C30705    JP 0507
342 REM        ××××××××××××××××××××××××××××××××××××××××
344 REM        39BF  3EFF      LD A,FF
346 REM        39C1  76        HALT
348 REM        39C2  D3FD      OUT(FD),A
350 REM        39C4  C9        RET
352 REM        ××××××××××××××××××××××××××××××××××××××××
354 REM        39D8  0698      LD B,98
356 REM        39DD  2D        DEC L
358 REM        39DE  C2D804    JP NZ,04D8
360 REM        39E1  06A2      LD B,A2
362 REM        39E3  C0        RET NZ
364 REM        39E4  25        DEC H
366 REM        39E5  F2D804    JP P,04D8
368 REM        39E8  C3E804    JP 04E8
370 REM        ××××××××××××××××××××××××××××××××××××××××
373 REM        39F2  10FE      DJNZ 39F2
374 REM        39F4  D2FE39    JP NC,39FE
376 REM        39F7  063B      LD B,3B
378 REM        39F9  063B      LD B,3B
380 REM        39FB  00        NOP
382 REM        39FC  10FE      DJNZ 39FC
384 REM        39FE  3E7F      LD A,7F
386 REM        3A00  DBFE      IN A,(FE)
388 REM        3A02  1F        RRA
390 REM        3A03  D0        RET NC
392 REM        3A04  0638      LD B,38
394 REM        3A06  AF        XOR A
396 REM        3A07  C32505    JP 0525
398 REM        3A0A  0100A3    LD BC,A300
400 REM        3A0D  F5        PUSH AF
402 REM        3A0E  0B        DEC BC
404 REM        3A0F  E3        EX(SP),HL
406 REM        3A10  E3        EX(SP),HL
408 REM        3A11  78        LD A,B
410 REM        3A12  B1        OR C
412 REM        3A13  20F9      JR NZ,3A0E
414 REM        3A15  F1        POP AF
416 REM        3A16  C9        RET
```

2. ábra



— a ZX81 és a használatos memóriabővítők a C000H—FFFFH tartományt legalábbis $\overline{MT}$ aktív állapotában a 4000H—7FFFH tartományra képezik le,

— a 8000H—FFFFH tartományból lehívott utasítás aktiválja az ULA videokontroller áramkörét, azaz

a) ha az utasítás 6-os bitje 0, a processzorra NOP utasítást kényszerít;

b) az $\overline{RFSH}$ jel alatt a ROM címsínjén az I regiszterből, a lehívott karakterkódból és egy belső számláló tartalmából összeállított cím jelenik meg. (A RAM címsínjén nem! Itt az I és R regiszter tartalma jelenik meg!);

c) ezalatt az ULA az éppen megcímzett memóriarekeszből olvasást hajt végre; ha ez ROM-rekesz, az ismertetett „kompozita" címről, ha RAM-rekesz, az I és

R adta címről. A tartalmat shiftregiszterbe teszi, és a következő NOP alatt min. 8 képpontot a képernyőre kiadja;

d) ha az ULA által kiolvasott utasítás 6-os bitje 1, ezt az utasítást eredeti alakban engedi a processzorhoz.

Programunk most már úgy fog működni, hogy képkiadáskor a processzort egy „becsapós" képernyősorra irányítjuk, mely 32 db NOP-ot tartalmaz, a képsor 32 karakterének megfelelően. Közben az I és R regiszter tartalmát úgy manipuláljuk, hogy a Spectrum képernyőfájl tartalma jelenjen meg a képernyőn. A becsapós sor 33. utasítása egy JP . . . utasítás, a következő sor kiadásának előkészítésére . . . és a folyamat így megy tovább, végig az összes soron.

A program az előzőek szerint egy NMI rutinból *(5. program)* és egy tulajdonképpeni képernyő-billentyűzet rutinból *(6. program)* áll.

Az NMI rutin hasonlóan működik a ZX81-éhez, de az AF' és IX regisztert meg akartuk menteni (ezeket a Spectrum használja), ezért a rendszerváltozó terület három nem használt helyét használtuk föl. Ez sajnos bizonyos időveszteséggel jár. Az NMI rutin a következő. Mivel az $\overline{\text{INT}}$-ot nem használjuk, az NMI rutin egy részét az itt levő Spectrum program egy része helyére írtuk.

Ha katalógusból kikeressük az egyes utasítások végrehajtásához szükséges időt, megállapíthatjuk, hogy minden sorkiadás összesen 207 ütemet vesz igénybe, ahogy ennek az előzőek értelmében lenni kell. Megjegyezzük, hogy az adódó kép a ZX81 képéhez képest kissé jobbra helyezkedik el; akit ez zavar, az a 38EBH—38EDH késleltetését csökkentheti (LD B, Ø2), ekkor a kép a ZX81-hez képest kissé balra lesz. Természetesen pontos egyezés is megvalósítható.

Mint látni fogjuk, a két „becsapós" sornak (a magyarázattól eltérően, ahol egyről beszéltünk, az idők „kisakkozása" miatt kettőre volt szükség) a kezdetét nem véletlenül választottuk 3880H-nak és 38CØH-nak. Így ugyanis könnyű olyan címértékelő áramkört készíteni, mellyel elérhető, hogy csak akkor kapcsoljuk be a videokontrollert, amikor ténylegesen képkiadás van. Ezzel az áramkörrel, valamint a RAM-nak $\overline{\text{MI}}$ aktív állapotától független címdekódolásával az egész RAM-ban futtathatunk gépi programokat.

Most már áttérhetünk a Spectrum ROM egyéb részeinek átalakítására. Ezek kevés kivételtől eltekintve a megszakításkezelés és a be-, kiviteli különbözőségek figyelembevételét jelentik.

Megjegyezzük, hogy az egyes programrészek eléggé összevissza helyezkednek el. Ennek oka, hogy az ismertetett megoldás számos próbálkozás eredménye; nem is célja a véglegesség, hanem az olvasó továbbfejlesztésre késztetése.

Ezek után lássuk a változtatásokat. Az érthetőség kedvéért helyenként változatlan részeket is megismételtünk (lásd a *7. programot*).

Miután az átalakításokat végrehajtottuk és a programot működésbe hoztuk, kipróbálhatjuk a Spectrum utasítások működését. Nyomtató hiányában nem foglalkoztunk a nyomtatót vezérlő utasításokkal. Nem módosítottuk a PAUSE utasítást sem. Ha mégis használni akarjuk, akkor értelemszerűen átalakítjuk, esetleg nagyobb számokat adunk meg. A gombok gyors megtalálásához a Spectrum-tasztatúrát a ZX81 billentyűzet fölötti üres helyre rajzoljuk. Célszerű a tápegység 5 V-os stabilizátorát is „kihelyezni".

Hardverkiegészítés

Ezután ismertetjük azt a hardverkiegészítést, mellyel elérhetjük, hogy az egész ØØØØH—FFFH tartományban futhassanak gépi programok. A kiegészítés egy 4 IC-s kapcsolás hozzáadásából áll, melyet például a tasztatúra alatt helyezhetünk el. A kapcsolás az *1. ábrán* látható.

Bekötéséhez az ULA 15-ös lábához menő fóliát át kell vágni, és a 157-es adatszelektor 4. lábát az ULA-hoz kötni. A működés azon alapszik, hogy ZX81-es üzemben az A15-öt az ULA A15 bemenetével kötjük össze, Spectrum üzemben pedig a 388Ø—389F és 38CØ—38DF címtartományok megszólításakor — ez a két pszeudo-képernyősor — az ULA A15 bementére H szintet kapcsolunk.

Most még a memóriakiegészítőt kell úgy átalakítanunk, hogy átkapcsolással a CØØØ—FFFF tartomány átkapcsolását $\overline{\text{MI}}$ aktív állapotában megszüntessük, az $\overline{\text{MI}}$-et ne engedjük érvényesülni. Ez a Memopak 64-nél is valószínűleg ugyanolyan jól fog működni, mint a szerző 32 k-sból 64 k-sra átalakított Audio Computers tárolójánál *(2. ábra)*.

Spectrum üzemben a kapcsolót zárjuk, az $\overline{\text{MI}}$ nem tud érvényesülni. A hardvermódosításokkal együtt a szoftvert is módosítani kell, ØØØ3H LD DE, B87F helyett LD DE,FFFF mert a Ramtopnál nem kell tekintettel lenni a pszeudo-képernyősorokra, azonfelül

```
39ØØ JP Pszeudo—1  +  8ØØØH helyett
                      JP Pszeudo—1
3912 JP Pszeudo—2  +  8ØØØH helyett
                      JP Pszeudo—2
```

mert a 15. címbit állítására nincs többé szükség a videokontroller működtetéséhez.

Megjegyezzük, hogy természetesen alkalmas helyre téve a „becsapós" képernyősorokat, az előző esetben is kiterjeszthetjük a memória felső korlátját majdnem FFFFH-ig.

Programmódosítás

A dolog izgalmasabb része most következik. Hogyan futnak a kész Spectrum programok az átalakított ZX81-en? Csak BASIC programokkal nincs különösebb baj. A gépi kódú programokhoz általában hozzá kell nyúlni. Egyszerűen sikerült adaptálni például az ismert GENS3M2—MONS3M2 programokat. Igaz, hogy csak a ZX81 FAST-hoz hasonlóan, azaz a kép gombnyomáskor megrándul, programvégrehajtás közben eltűnik. Sokat ígérő megoldás lehet az, hogy az $\overline{\text{NMI}}$ helyett az $\overline{\text{INT}}$-ot használjuk. Ez további IC-k beépítésével és az üzemi rendszer hasonló elveken alapuló átalakításával megoldható. Így a gépi kódú programokba még kevésbé kell belenyúlnunk. Ennek végrehajtását és egyéb programok futásképessé tételét az érdeklődő — és kitartó — olvasóra bízzuk.

A hivatkozott szakirodalom

(1) Memopak-Riesen wachgerüttelt. c't, 1985/4.

(2) Für alle Mikrocomputer: Kasettenaufzeichnung mit 3600/7200 Baud. Super-Tape mit dem Z80 am Beispiel des ZX81—16K. c't, 1984/4.
Super-Tape Komfort für den ZX81. c't,1984/5.

(3) Digitaler Kasettenrekorder mit ZX81. Elektor, 1984. június

(4) Digitaler Kasettenrekorder. Elektor, 1984. január

(5) Tempo durch Nachbrenner. Funkschau, 1984/4.

(6) Die Trickreiche ZX81 Videoausgabe. c't, 1985/4.

(7) ZX-Spectrum ROM programja. 1985. Ipari Informatikai Központ

(8) ZX-Spectrum. Bevezető és BASIC programozási kézikönyv. IIK, 1985.

(9) ZX-Spectrum. BASIC és gépi kódú programozás. IIK, 1985.

(10) Spectrum Hardware Manual. Melbourne House Publishers.

(11) ZX81 Kochbuch. Funkschau Sonderheft Nr.8. 1984.

(12) Logan—O'Hara: The Complete Timex TS1000/Sinclair ZX81 ROM Disassembly. Melbourne House Publishers.

BEKE GYULA

Közületek, figyelem!

Mikroszámítógépet akarnak vásárolni?

Tájékozódjanak előtte a naprakész piaci helyzetről!

Díjtalan ismertető:

MESZ Számítástechnika 1368 Budapest Pf. 193.

μINFORM

Kedves diákok és tisztelt deákok! A mit, mivel, mikor, hol és hogyan kezdetű kérdésekre a mikroszámítógépes szakirodalom kapcsán sem könnyű felelni...E kérdések megválaszolása attól is függ, hogy ki és mennyire ismeri a nyelveket, amelyek az elsődleges forrásokból való merítéshez csobolyóul szolgálnak. Pedig bizony dús hozamúak e források: a külföldről könyvtárainkba rendszeresen beáramló mikroszámítógépes lapok.

E szaklapok között még olyanok is akadnak, amelyekből - magáncélra - kimásolhatók a közölt programlisták, és amelyekben bővítő/illesztőkártyák vagy más egységek kapcsolási rajzai vannak. Ezek hasznosításához még nyelvtudás sem kell! (Nyelvet tanulni, kedves diákok, azért - kell !)

A nyelveket ismerő tisztelt olvasóink közül bizonyára többeknek a "mikor" a legnagyobb gondjuk. Nekik is segíthetünk talán azzal, hogy e források feltárásában az első fázisokat készen kínáljuk. Így - már megnyerve az előválogatás idejét - több lehetőségük marad a számukra valóban érdekes programozási fogások, üzemeltetési, karbantartási tanácsok és a - még a sovány magyar túristapénztárcával is megszerezhető - perifériaújdonságok megismerésére, továbbá olyan cikksorozatok figyelemmel kísérésére, amelyek igazán remekül vannak szerkesztve és célzottan informatívak.

Új szolgáltatást vezetünk be, ami által elsősorban Apple, Atari, Commodore és Sinclair modellek tulajdonosainak a mindennapi munkáját/szórakozását/tanulását reméljük megkönnyíteni. Minden számunkban tartalomleírást teszünk közzé ezután a következő folyóiratok aktuális közleményeiről:.

A folyóirat neve	Azonosító kódja
64'er Magazin	64 er
Chip Magazin	chip
Commodore Horizons	coho
Commodore Microcomputers	comi
Compute!	cute
Computer Persönlich	pers
Happy Computer	happ
hc - Mein Home-Computer	hc
mc - Zeitschrift	mc
Run /USA/	run
Sinclair User	sinc
Your Sinclair	ysin

Első lépésként a lista szerinti folyóiratok 1986.évi első számait dolgoztuk fel C-64/1541/802 rendszerrel olymódon, hogy a tartalomleíró szóláncokat permutáltuk, a szóláncvariánsokat pedig alfabetikusan rendeztük. Így lehetőségünk nyílik arra, hogy olvasóink kívánsága szerint válasszuk ki a témákat: pl. "ATARI" vagy "barkácsolás", vagy akár "cikksorozat", "termékismertetés", stb... Most, bemutatásképpen, az adatbázis "programlista" kezdetű rekordjainak listáját mellékeljük.

A közölt listából az érdeklődők készen kapják tehát a legfrissebb információkat, vagyis hogy kedvenc gépükhöz vagy érdeklődési körükhöz kapcsolódóan mi jelent meg ezekben az újságokban. Így már csak a "hogyan szerezzem meg?" kérdés marad: másolatot lehet kérni a könyvtáraktól a kiválasztott anyagról.

(A folyóiratok például a SZÁMALK szakkönyvtárában föllelhetők. A másolás feltételeiről bővebb felvilágosítást a 853-111/251 telefonszámon kaphatnak olvasóink.)

A tartalomleírás egy szóláncból áll, majd a listában ezt követi a forrás megjelölése a folyóirat azonosítójával, a megjelenés dátumával és a cikk előkereséséhez a kezdő oldalszám és a terjedelem megadásával. A mellékelt lista értelmezéséhez még az alábbiakat kell tudni. A tartalomleírás szóláncában elsőként a téma átfogó megnevezése, utána a számítógéptípus(ok), ezt követően a szűkebben jelölt tartalom meghatározása szerepel, majd esetlegesen néhány, a közleményt minősítő adat (például : cikksorozat).

A forráshely karaktersorozatát nyíl vezeti be, melyet a / jelig a folyóiratok azonosítója, a két / jel között az évszám, folyóiratszám és kötőjellel a kezdő oldalszám követi, a végén pedig a közlemény teljes oldalterjedelme áll.

Reméljük, hogy olvasóink - a diákok és deákok egyaránt - az első listánkat tanulmányozva máris találnak érdekességeket, amelyek hasznosításához sok sikert kívánunk.

Van még egy javaslatunk : Európa legjobb Commodore újságjáról, a 64'er Magazinról másféle információszolgáltatás is kérhető. Szolnoki Béla vállalja, hogy havi 20 Ft-ért (+ a postaköltség) elküldi a folyóirat tartalomjegyzékeinek megjegyzésekkel is ellátott fordítását - a lap 1986. 07. havi számától kezdődően bármelyikről, megbízás esetén folyamatosan. Az eligazító tartalomjegyzék alapján a megrendelő kiválaszthatja, hogy számára melyik cikkek fontosak, és ezek fordítását is megkaphatja - oldalanként 5 Ft-ért (+ a postaköltség). Különös aktualitását adhatja a megrendelésnek, hogy a 64'er Magazin 1986.októberétől sorozatot indított kezdő C64-eseknek!

Összehasonlítva az árakat bármely hasonló szolgáltatással, kedvező az eredmény, ezért **ajánljuk olvasóinknak a "Számítástechnikai információszolgáltatás"-t Szolnoki Béla, 1446 Budapest, Pf 400. címen.**

A Szerkesztőség

```
MUVELETGYORSITAS
   programlista commodore 64 lemezegyseg (1541) <hypra-load> <ultraload plus> osszehasonlito ertekeles
   ->happ/86.01-48/5
MUVELETGYORSITAS
   termekismertetes commodore 64 lemezegyseg (1541) <turbotrans> <turbo-access> ->chip/86.01-66/1
OKTATOPROGRAM KESZITES
   programozasi tanacsok commodore 64;128 data-programsorok
   ->comi/86.01-58/2
PASCAL PROGRAMOZAS
   programlista hexload program <turbo-pascal> ->pers/86.01-42/2
PASCAL PROGRAMOZAS
   termekismertetes compiler kinalatos szehasonlito ertekeles
   ->pers/86.02-100/2
PL/1 PROGRAMOZAS
   programlista cikksorozat assembler rutin behivas ->chip/86.01-152/3
PL/1 PROGRAMOZAS
   programlista linearis optimalas modszertan absztrakt pelda
   ->pers/86.01-50/8
PROGRAMLISTA
   adatallomany kezeles commodore 64 basic bovites <hypra-load> <datawork-basic> ->64er/86.01-54/5
```

PROGRAMLISTA
adatallomany vedelem commodore 64 rejtjelezes vedoprogram
->comi/86.01-54/3
PROGRAMLISTA
adatatvitel apple ii cp/m lemezformatum illesztes ->mc/86.01-62/5
PROGRAMLISTA
apple ii atari 400/800/xl/xe commodore 16/64/128/plus/4 jatekprogram (paszianSz) ->cute/86.01-48/7
PROGRAMLISTA
apple ii grafikus programozas 3d-grafika ->hc/86.01-67/3
PROGRAMLISTA
apple macintosh memoriatartalom kiiratas (ramedit) ->pers/86.01-46/2
PROGRAMLISTA
apple billentyuzet modositas
->cute/86.01-102/2
PROGRAMLISTA
apple ucsd-pascal karaktertipus-valtas ->mc/86.01-86/1
PROGRAMLISTA
assembler programozas commodore 64/128 (es-ae 64) alkalmazasi utmutato
->happ/86.01-56/7
PROGRAMLISTA
atari 130xe (copy 130) egymenetes lemezmasolo ->happ/86.01-86/3
PROGRAMLISTA
atari 400/800/xl/xe adatvedo rutin reset/break kiiktatas
->cute/86.01-109/2
PROGRAMLISTA
atari 400/800/xl/xe basic programozas sortorlo rutin ->cute/86.01-112/1
PROGRAMLISTA
atari 400/800/xl/xe verifikalas (Kazetta) ->cute/86.01-100/1
PROGRAMLISTA
atari 520st (scopy 520 st) filemasolas ->happ/86.01-83/2
PROGRAMLISTA
atari 600/800/xl jatekprogram (dungeons of xotha) ->hc/86.01-70/5
PROGRAMLISTA
atari 800xl listazas-lassitas (slowlist) ->hc/86.01-94/2
PROGRAMLISTA
barkacsolas commodore 64 illeszto digitalizalo vezerles
->run/86.01-35/4
PROGRAMLISTA
barkacsolas commodore 64 lemezegyseg (1541) muveletgyorsitas busz (parhuzamos) ieee-dos generalas
->64er/86.01-144/3
PROGRAMLISTA
basic programozas commodore 64 lista-formatalas ->run/86.01-74/3
PROGRAMLISTA
basic programozas commodore 64/128 (error 64) hibas programsor kijelzese
->happ/86.01-74/1
PROGRAMLISTA
cikksorozat commodore 64 muveletgyorsitas gepikod generalas (ascompiler-64) ->64er/86.01-58/7
PROGRAMLISTA
cikksorozat commodore 64/128 grafikus programozas koordinata transzformacio hires-3 ->64er/86.01-131/5
PROGRAMLISTA
cikksorozat pl/1 programozas assembler rutin behivas ->chip/86.01-152/3
PROGRAMLISTA
commodore 128 sprite-kezeles
->comi/86.01-90/3
PROGRAMLISTA
commodore 64 (disassembler 64)
->cute/86.01-89/2
PROGRAMLISTA
commodore 64 (scroll-64) programlista-gorgetes ->happ/86.01-74/2
PROGRAMLISTA
commodore 64 (speedscript) modositas
->cute/86.01-10/1
PROGRAMLISTA
commodore 64 (speedscript) karaktertipus modositas ->cute/86.01-107/3
PROGRAMLISTA
commodore 64 grafikus programozas animacios program (pulsing pictures)
->run/86.01-48/5
PROGRAMLISTA
commodore 64 hangjelzes (vesszo/return eseten) ->pers/86.02-46/1
PROGRAMLISTA
commodore 64 hex-billentyuzet
->64er/86.01-78/1
PROGRAMLISTA
commodore 64 jatekprogram (magikus mezo) ->hc/86.01-43/3
PROGRAMLISTA
commodore 64 jatekprogram (word jumbler) ->run/86.01-104/3
PROGRAMLISTA
commodore 64 kurzorvezerles fuzerbeiras (control-q) ->comi/86.01-52/1
PROGRAMLISTA
commodore 64 lemezegyseg (1541) lemeztar karbantartas masoloprogram
->coho/86.01-49/2
PROGRAMLISTA
commodore 64 lemezegyseg (1541) muveletgyorsitas (hypra-load) (ultraload plus) osszehasonlito ertekeles
->happ/86.01-48/5
PROGRAMLISTA
commodore 64 list-kiiktatas
->hc/86.01-85/4

PROGRAMLISTA
helyesbites (cute/85.11-92)
->cute/86.01-87/1
PROGRAMLISTA
helyesbites (hc/85.09-44)
->hc/86.01-22/1
PROGRAMLISTA
KeresKedelem mbasic programozas KalK
ulacios programoK ->pers/86.01-44/3
PROGRAMLISTA
linearis optimalas pl/1 programozas
modszertan absztraKt pelda
->pers/86.01-50/8
PROGRAMLISTA
mbasic programozas Kepernyoszoveg Ki
iratas ->pers/86.02-44/1
PROGRAMLISTA
pascal programozas hexload program (
turbo-pascal) ->pers/86.01-42/2
PROGRAMLISTA
sinclair ql lemezformatalas filemaso
las ->pers/86.02-46/2
PROGRAMLISTA
sinclair spectrum (bunte pause) Kere
tszinezes ->happ/86.01-89/1
PROGRAMLISTA
sinclair spectrum ablaKKijeloles
->sinc/86.01-62/3
PROGRAMLISTA
sinclair spectrum jateKprogram (adve
nturer) ->sinc/86.01-107/3
PROGRAMLISTA
sinclair spectrum jateKprogram (chop
per mission) ->ysin/86.01-58/4
PROGRAMLISTA
sinclair spectrum jateKprogram (droi
d wars) ->ysin/86.01-101/4
PROGRAMLISTA
sinclair spectrum jateKprogram (KraK
atoa) ->sinc/86.01-104/3
PROGRAMLISTA
sinclair spectrum jateKprogram (towe
r of doors) ->sinc/86.01-101/2
PROGRAMLISTA
sinclair spectrum jateKprogram (wash
ing line) ->sinc/86.01-103/1
PROGRAMLISTA
sinclair spectrum jateKprogram (worm
) ->ysin/86.01-62/2
PROGRAMLISTA
sinclair spectrum jateKprogram jateK
automata szimulator
->hc/86.01-54/10
PROGRAMLISTA
sinclair spectrum monitorKep tarolas
->sinc/86.01-71/2
PROGRAMLISTA
sinclair spectrum morse-Kod generala
s ->ysin/86.01-61/2
PROGRAMLISTA
commodore 64 nem-garantalt lemezhato
ldal vizsgalat ->hc/86.01-85/1
PROGRAMLISTA
commodore 64 nyomtato rutin Kepernyo
szoveg Kiiratas ->64er/86.01-64/5
PROGRAMLISTA
commodore 64 oraido megjelenites
->64er/86.01-78/2
PROGRAMLISTA
commodore 64 uzenetmegjelenites szov
egmozgato rutin ->cute/86.01-111/1
PROGRAMLISTA
commodore 64;128 (speedcalc) alKalma
zasi utmutato ->cute/86.01-66/12
PROGRAMLISTA
commodore 64;128 grafiKus programoza
s sprite-vezerles ->happ/86.01-73/1
PROGRAMLISTA
commodore 64;128 jateKprogram (taxi)
felhasznalasi utmutato
->happ/86.01-68/5
PROGRAMLISTA
commodore 64;128 tarKapacitas Kijelz
es (free 64) ->happ/86.01-73/1
PROGRAMLISTA
commodore 64;vc20 list-KiiKtatas (au
to-run) ->run/86.01-100/2
PROGRAMLISTA
commodore 16;128;plus/4 gepiKod viss
zaalaKitas (datastater) basic progra
m gepiKodbol ->comi/86.01-61/3
PROGRAMLISTA
commodore 16;64;plus/4;vc20 nyomtato
program listazas ->cute/86.01-99/2
PROGRAMLISTA
cp/m fileKezeles (unera) fileregener
alas ->pers/86.01-49/1
PROGRAMLISTA
cp/m mbasic programozas fileKezeles(
random) ->pers/86.02-48/3
PROGRAMLISTA
dbase ii nyomtato alprogram
->pers/86.02-52/5
PROGRAMLISTA
dbase ii ragaszthato cimKeK nyomtata
sa ->pers/86.02-45/1
PROGRAMLISTA
hc-mein home-computer beviteli utmut
ato atari commodore ->hc/86.01-75/1
PROGRAMLISTA
helyesbites (chip/85.09-132)
->chip/86.01-21/1
PROGRAMLISTA
helyesbites (chip/85.10-188)
->chip/86.01-21/1
PROGRAMLISTA
helyesbites (cute/85.11-125)
->cute/86.01-87/1

A Beta Basic és alkalmazása III.

A szótárprogram magyar ábécés változata

A program a magyar ékezetes betűket grafikus üzemmódban írja ki. Emiatt a Beta basicet KEYWORDS Ø üzemmódba kellett állítani (G kurzor 8). Ebben az üzemmódban a Beta zavartalanul működik, kivéve az ON ERROR utasítást. Grafikus üzemben az új kulcsszavak helyett a felhasználói grafikus karakterek íródnak ki a képernyőre. Ez a jelenség a program listázásánál is megfigyelhető.

Például az 55-ös sor:

```
KEYW. Ø:
55 PRINT # Ø;AT 1,1; „Törléshez nyomd
        meg az ENTER-t!"
KEYW. 1:
55          ; „T DPOKE rl ROLL shez
        nyomd meg az ENTER-t!"
```

Vagy a 45-ös sor:

```
KEYW. Ø:
   45 GO TO óa
KEYW. 1:
   45 GO TO ON a
```

Lehet, hogy ez a programlista kicsit furcsán hat, de a program futása ettől még zavartalan. Mivel az ON ERROR így nem használható, az egész programból ki kell törölni (15, 215, 315, 410, 730, 810-es sorok). Ugyanez vonatkozik a POP utasításokra is (220, 815-ös sorok). Mivel az ON ERROR-t elhagytuk, nincs szükség a program futásának megállítására sem. Ezért a lista eleje a következő lesz:

```
1 GO TO 3
2 PRINT AT 17,Ø: LOAD "" CODE
3 INK Ø: DELETE 1 TO 3
```

Természetesen SAVE "pr." LINE 2. Ekkor a program megszakítás nélkül fut, de az első, például magnótöltési hibánál hibaüzenet helyett csak villogó K kurzort látunk. A 4-5-ös sor elmarad. A 20-as sorban beállítjuk a KEYW. Ø-t.

```
2Ø KEYWORDS Ø : PRINT AT 1,1Ø;"
S Z Ó T Á R"
```

Már csak azt kell megoldanunk, hogy a scrollozás zavartalan legyen. Ez nekem a következőképpen sikerült. Felvettem egy scr nevű változót, amit a kereső eljárás meghívása előtt alaphelyzetbe állítok:

```
212 LET scr=1
215 PROC kereses
```

valamint

```
81Ø PRINT INVERSE 1;k$.LET scr=
1.PROC szotorles
```

Ezután a kereső eljárást kell módosítani.

```
1157 LET scr=scr+INT (LEN i$/32)+1;
```

vagyis számoljuk a sorokat.

Én egy képernyőre 19 sort írattam, mivel az előző változatnál a felső két sor mindig elveszett a 2ØØ INPUT sor miatt. Vagyis:

```
1158 IF  scr>=2Ø  THEN  POKE
     23692,255:  PRINT#Ø,AT  2,Ø;
     "scroll?":
     GET a$: IF a$="n" OR a$="N"
     OR a$=CHR$ 32 THEN LET h=Ø:
     GO TO 1175: ELSE LET scr=1:
     PRINT#Ø,AT 2,Ø;"          ":
     GO TO 116Ø
```

Tehát ha a sorok száma elérné a 20-at, a 23692-es rendszerváltozóba 255-öt írva kikapcsolom a gép saját scrollozó rendszerét, és az enyém lép működésbe. A szokott módon felteszi a "scroll?" kérdést, és "n" vagy SPACE válaszra az eljárás végére ugrik, be-

---

```
PROGRAMLISTA
   sinclair spectrum tarKapacitas Kiira
   tas ->hc/86.01-94/1
PROGRAMLISTA
   sinclair user beviteli utmutato
   ->sinc/86.01-101/1
PROGRAMLISTA
   termeKvalasztek a/d atalaKito commod
   ore 64 alKalmazasi pelda
   ->comi/86.01-66/6
PROGRAMLISTA
   terminal uzemmod ciKKsorozat commodo
   re 64 (runterm plus)
   ->run/86.01-84/7
PROGRAMLISTA
   uzemeltetesi tanacsoK commodore 64 l
   emezegyseg(1541) reset-szoftver
   ->cute/86.01-20/1
PROGRAMLISTA
   veletlenszam generalas gauss fele ve
   letlenszam ->happ/86.01-94/2
PROGRAMLISTA-BEVITELI MOD
   commodore microcomputers
   ->comi/86.01-123/3
PROGRAMOZASI TANACSOK
   commodore 64 felhasznaloi javaslatoK
   ->64er/86.01-74/4
PROGRAMOZASI TANACSOK
   commodore 64;128 oKtatoprogram Keszi
   tes data-programsoroK
   ->comi/86.01-58/2
PROGRAMOZASI TANACSOK
   dbase ii alKalmazasi utmutato
   ->pers/86.01-41/1
PROGRAMOZASI TANACSOK
   sinclair spectrum felhasznaloi javas
   latoK ->ysin/86.01-29/1
PROLOG NYELV
   adatbazis leKerdezes ciKKsorozat log
   iKai programozas ->pers/86.01-58/2
PROLOG NYELV
   ciKKsorozat logiKai programozas reKu
   rziv hivas ->pers/86.02-64/3
PROLOG PROGRAMOZAS
   termeKismertetes commodore 64 (micro
   -prolog) ->coho/86.01-41/1
ROBOTRON(NDK)
   termeKismertetes matrixnyomtato(lq)
   teszteles illesztesi opcioK
   ->pers/86.01-21/3
ROM-RUTINOK
   ciKKsorozat commodore 64
   ->64er/86.01-92/2
SINCLAIR QL
   programlista lemezformatalas filemas
   olas ->pers/86.02-46/2
SINCLAIR SPECTRUM
   jateKprogram oroKelet poKe-lista
   ->ysin/86.01-28/2
```

fejezve a kiírást. Ha itt nem állítanám Ø-ra a h változót, akkor szótörlésnél esetleg olyan szót is kitörölhetnék, ami még nem is jelent meg a képernyőn, hiszen a kiírás ezután következik a 1160-as sorban. Más billentyűt lenyomva folytatódik a kiírás.

A sok feltételes elágazás miatt a program futása kicsit lelassult, ezért az 1160-as sort módosítottam: 1160 PRINT i$: PAUSE 10

```
   2 CLEAR 28699: GO TO 4
   3 PLOT 25,140: DRAW 200,0: DR
AW 0,-110: DRAW -200,0: DRAW 0,1
10: PRINT AT 5,7;"Litauszky  Gyo
rgy";AT 10,7;"** S Z O T A R **"
;AT 16,12;"© 1986": RETURN
   4 GO SUB 3: INK 7: PRINT AT 1
7,0: LOAD ""CODE : RANDOMIZE USR
 58419
   5 CLS : INK 0: GO SUB 3: INK
7: PRINT AT 17,0: POKE 23609,5
   6 LOAD ""
```

```
Litauszky  Gyorgy


** S Z O T A R **


© 1986
```

A loader:

```
   2 CLEAR 28699: GO TO 4
   3 PLOT 25,140: DRAW 200,0: DR
AW 0,-110: DRAW -200,0: DRAW 0,1
10: PRINT AT 5,7;"Litauszky  Gyo
rgy";AT 10,7;"** S Z O T A R **"
;AT 16,12;"© 1986": RETURN
   4 GO SUB 3: INK 7: PRINT AT 1
7,0: LOAD ""CODE : RANDOMIZE USR
 58419
   5 CLS : INK 0: GO SUB 3: INK
7: PRINT AT 17,0: POKE 23609,5
   6 LOAD ""
```

A menu

```
S Z O T A R


   1.   Beiras
   2.   Kereses
   3.   Kimentes
   4.   Betoltes
   5.   Szo torlese
   6.   File torles
   7.   Informacio


        © 1986
```

Kereses:

```
air
affair=ugy
air mail=legiposta
air=levego
aircraft=repulogep
air travel=legi utazas
airliner=utasszallito
airport=repuloter
airplane=repulogep
airscrew=legcsavar
airframe=sarkany(rep.gepvaz)
airfield=repuloter
airborne=legi uton szallitott
chair=szek
fairly=majdnem,szepen,megtheto-
sen
pair=par,parosodik


KERESES?          Visszateres:0
■
```

Ha tenyleg csak az air-tkeressuk

```
air
air mail=legiposta
air=levego
aircraft=repulogep
air travel=legi utazas
airliner=utasszallito
airport=repuloter
airplane=repulogep
airscrew=legcsavar
airframe=sarkany(rep.gepvaz)
airfield=repuloter
airborne=legi uton szallitott
KERESES?          Visszateres:0
■
```

Informacio

```
      I N F O R M A C I O

A tar merete:          27060 byte
Felhasznalva:          16952 byte
Szabad:                10108 byte
Beirva:                  789 szopar
```

```
28700                          55760


   Nyomj meg egy billentyut!
```

A program:

```
   1>GO TO 5
   2 PRINT AT 17,0: LOAD ""CODE
   3 INK 0: PRINT BRIGHT 1;AT 10
,4;"Indits el a programot.";AT
11,10;"(RUN+ENTER)"
   4 STOP
   5 DELETE 1 TO 5
  10 CLEAR
  15 ON ERROR 15
  20 PRINT AT 1,10;"S Z O T A R"
  25 PRINT AT 2,2;"
  30 PRINT AT 5,7;"1.   Beiras";
AT 7,7;"2.   Kereses";AT 9,7;"3.
  Kimentes";AT 11,7;"4.   Betol
tes";AT 13,7;"5.   Szo torlese";
AT 15,7;"6.   File torles";AT 17
,7;"7.   Informacio"
  35 PRINT AT 20,2;"
  40 PRINT #0;AT 0,11;"©  1986"
: GET a
  45 CLS : GO TO ON a;100,200,30
0,400,800,55,700
  50 GO TO 10
  55 PRINT #0;AT 1,1;"Torleshez
nyomd meg az ENTER-t!": PAUSE 0
  60 LET r$=INKEY$: IF r$<>CHR$
13 THEN GO TO 70
  65 CLEAR : GO SUB 500: PRINT A
T 19,8;"FILE TAROLVA!": GO TO 15
  70 GO TO 10
 100 INPUT AT 0,0;"BEIRAS";TAB
16;" Visszateres:0";AT 2,0; LIN
E b$
 105 IF b$="0" THEN GO TO 10
 110 PROC beiras
 115 PRINT b$
 120 GO TO 100
 200 INPUT AT 1,0;"KERESES";AT
1,16;" Visszateres:0";AT 3,0; L
INE k$
 205 IF k$="0" THEN GO TO 10
 210 PRINT INVERSE 1;k$
 215 ON ERROR 200: PROC kereses
 220 POP : GO TO 200
 300 INPUT AT 0,0;"KIMENTES";T
AB 15;"Filenev:";AT 2,0;m$
 305 GO SUB 600
 310 SAVE m$CODE 28700,cim-28700
 315 ON ERROR 300
 320 INPUT "VERIFY (i/n)";v$
 325 IF v$="n" THEN GO TO 10
 330 VERIFY m$CODE
 335 GO TO 10
 400 GO SUB 600: IF cim<>28700 T
HEN PRINT AT 10,1;"Betoltes  elo
tt  a  szoallomanytorlesehez  ha
sznald  a  menu 6.";TAB 12;"opci
ojat.": PRINT #0;AT 1,3;"Nyomj m
eg egy billentyut!": PAUSE 0: GO
 TO 10
 405 INPUT AT 0,0;"BETOLTES ";
TAB 15;"Filenev:";AT 2,0;m$
 410 ON ERROR 400
 415 LOAD m$CODE
 420 GO TO 10
 500 RANDOMIZE USR 55783
 505 RETURN
 600 LET cim=INSTRING(28700,MEMO
RY$(8 TO 8))
 605 RETURN
 700 GO SUB 600: PRINT AT 1,6;"I
N F O R M A C I O";AT 5,1;"A t
ar merete:";TAB 20;"27060 byte";
AT 7,1;"Felhasznalva:";TAB 20;c
im-28700;" byte";AT 9,1;"Szabad:
";TAB 20;55760-cim;" byte"
 705 PLOT 0,50: DRAW 255,0: DRAU
0,5: DRAW -255,0: DRAW 0,-5
 710 PLOT 0,50: DRAW 0,-4: PLOT
128,50: DRAW 0,-3: PLOT 255,50:
DRAW 0,-4
 715 PLOT 0,52: DRAW (cim-28700)
/106,0: DRAW 0,1: DRAW -(cim-287
00)/106,0
 720 PLOT 0,40,"28700": PLOT 215
,40,"55760"
 725 LET a=USR 55760: PRINT AT 1
1,1;"Beirva:";TAB 20;a;" szopar"
 730 ON ERROR 200
 735 PRINT #0;AT 1,3;"Nyomj meg
egy billentyut!": PAUSE 0: GO TO
 10
 800 INPUT AT 1,0;"SZOTORLES";
TAB 16;"Visszateres:0";AT 3,0; L
INE k$
 805 IF k$="0" THEN GO TO 10
 810 PRINT INVERSE 1;k$: ON ERRO
R 800: PROC szotorles
 815 POP : GO TO 800
1000 DEF PROC beiras
1005 GO SUB 600
1010 IF 55759-cim<=LEN b$+1 THEN
 PRINT "A szotar megtelt": GO T
O 200
1015 POKE cim,42
1020 POKE cim+1,b$
1025 END PROC
1100 DEF PROC kereses
1105 LET c=INSTRING(28700,MEMORY
$() TO 55760),k$)
1110 IF c=0 THEN PRINT "Nem isme
rek ilyen szot.": LET h=0: GO TO
 1175
1115 LET h=1
1120 DO
1125 EXIT IF PEEK c=42
1130 LET c=c-1
1135 LOOP
1140 LET c1=c
1145 LET c2=INSTRING(c+1,MEMORY$
() TO 55760),CHR$ 42)
1150 IF c2=0 THEN GO SUB 600: LE
T c2=cim
1155 LET i$=MEMORY$()(c1+1 TO c2
-1)
1160 PRINT i$: PAUSE 20
1165 LET c=INSTRING(c2,MEMORY$()
( TO 55760),k$)
1170 IF c<>0 THEN GO TO 1120
1175 END PROC
1200 DEF PROC szotorles
1205 PROC kereses: IF h=0 THEN G
O TO 1245
1210 PRINT #0,AT 1,3,"Az utolso
szo torlese: ^"1"^^: GET a
1215 GO TO ON a;1225
1220 GO TO 1245
1225 GO SUB 600: POKE c1,STRING$
(LEN i$+1,CHR$ 0)
1230 POKE c1,MEMORY$()(c2 TO cim
)
1235 GO SUB 600: POKE cim+1,STRI
NG$(LEN i$,CHR$ 0)
1240 PRINT "Az utolso szo torolv
ve!"
1245 END PROC
```

Meg egy kis kereses:

```
wing
wing=szarny
wing flap=fekszarny
wingspain=fesztav
wing area=szarnyfelulet
raise
wing=szarny
wing flap=fekszarny
wing area=szarnyfelulet
novel
increase=novel,szaporit,fokoz,e-
mel
raise=emel,novel,ebreszt,okoz
write
raise=emel,novel,ebreszt,okoz


KERESES?          Visszateres:0
■
```

Szükség volt még arra is, hogy a kiírást bármikor meg lehessen állítani. Erre a SPACE billentyűt választottam.

1162 IF INKEY$=CHR$ 32 THEN GO TO 1175 ; elj. végére ugrik.

Ezután a Spectrum-kézikönyv utasításai szerint elkészíthetjük a magyar ékezetes betűket, és hozzáfűzhetjük a Beta gépi kódú programrészéhez. Mivel szótörlésnél „az utolsó szó törölve!" felirat nem elég egyértelmű, az 1240-es sort így módosítottam:

1240 PAPER 5: PRINT i$;TAB 31;" ": PAPER 7: PRINT " ";TAB 11;"TÖRÖLVE"; TAB 31;" "

A PAPER 7 utáni két " " és a "TÖRÖLVE" felirat kék PAPER színvezérléssel van beírva.

Természetesen még sok más dolog megoldható lenne a programmal, de minél tovább bővítenénk, annál feljebb kellene helyezni a RAMTOP-ot, ez pedig a beírható szavak számát csökkentené.

Például ha átjavítjuk az 1210-es sort a következőképpen: 1210 PRINT#0,AT 0,0; i$; TAB 31;" ": PRINT 0;" "; TAB 10; "TÖRLÉSE: ""1""; TAB 31;" ": GET a akkor szótörlésnél törlés előtt is kiírja a gép a törlendő szót. Ekkor azonban ha a RAMTOP-ot nem helyezzük feljebb, akkor már csak kb. 5 sor terjedelmű bejegyzést írhatunk be, mert ha ki akarnánk törölni, akkor a „4 Out of memory" hibaüzenetet kapjuk. Mindenesetre a program szótárnak még így is használható. Én szavak beírásánál egyszer sem használtam fel két sornál többet.

Még egyszer a Beta basicról

Ha valakinek megvan a Beta programja, de nem ismeri, írja be a következő rövid programot a gépbe:

10 FOR n= 255 TO 0 STEP −1
20 PLOT n,88;"Fenyujsag"
30 NEXT n
40 GO TO 10

Ez a lényegében még csak Spectrum basic kulcsszavakat tartalmazó, de már Beta basicben írt programrész egy balra scrollozó gépi kódú rutint helyettesít. A szöveg jobbról balra átúszik a képernyőn, méghozzá úgy, hogy ami eltűnik a bal oldalon, az újra megjelenik jobbról.

Már ebből is látható, hogy milyen új lehetőségeket nyújt a Beta. Az új ROLL és SCROLL kulcsszavak pedig ennél sokkal többre képesek.

Sokkal könnyebb a programírás az új programszerkesztő parancsokkal. Ezek a következők:

EDIT (shift nélküli 0, de csak ENTER után)

Például: EDIT 210 ENTER.

Az ENTER-t lenyomva megjelenik a 210-es sor az alsó képernyőn, és máris javítható. A javítandó sorban mind a négy(!) kurzorgomb értelemszerűen használható. Ez még nem fullscreen editor, de már nem sok hija.

RENUM (G kurz. 4)

Például: RENUM + ENT : Az egész programot átsorszámozza, az első új sorszám a 10, a növekmény 10.

Az átsorszámozó rutin a képernyő-memóriát használja, ezért átsorszámozás közben csíkokat láthatunk a képen, de ezek az ENT lenyomása után eltűnnek. A rutin átjavítja a GO TO és GO SUB utasításokat is, de NEM javítja át a GO TO ON utasítás után álló értéket.

DELETE (G kurz. 7)

Erről már volt szó.

AUTO (G kurz. 6)

Például AUTO ENT: automata sorszámadás a következő sortól, tízesével növekedve.

AUTO 25 ENT: automata sorszámadás 25-től, tízesével növekedve.

AUTO 30,4 ENT: automata sorszámadás 30-tól négyesével növekedve.

Hiba: NEW után AUTO parancsra a sorszámadás nem 10-től, hanem az előző (kitörölt) program aktuális sorától kezdődik.

JOIN (G kurz. SYM SH+6)

Az aktuális sort (>jel) összeolvasztja az utána következővel. Az új sorszám az aktuális soré lesz, a két előző programsort értelemszerűen kettőspont választja el egymástól, és a magasabb sorszám megszűnik.

SPLIT

Ez nem kulcsszó, hanem normál, tehát nem grafikus kurzorral beírt „nem egyenlő" jel (SYM SH+W).

Ha egy az alsó képernyőn lévő sor bármelyik, de nem első utasítása elé beírjuk a <> jelet, az ENTER-t lenyomva a <> jel előtti rész felkerül a felső képernyőre, a maradék azonos sorszámmal a szerkesztőmezőben marad. Így ha ennek sorszámát átjavítjuk, két külön sort kapunk.

Helyhiány miatt már csak a **DEF KEY** utasítást említem meg (G. kurz. SYM+1). Ezzel bármelyik szám- vagy betűbillentyűhöz tetszőleges feladatot rendelhetünk, és így 36 db(!) funkcióbillentyűhöz jutunk.

Írjuk be például:

DEF KEY "m": PRINT FN m//;" bytes free" +ENT

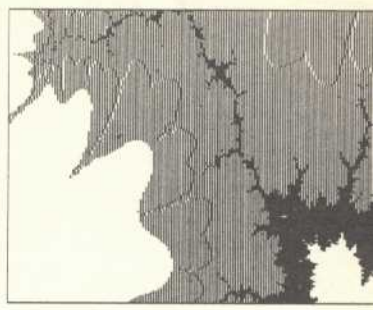
A válasz "0 OK,0:1". Ha most megnyomjuk a DEF KEY shiftet (SYM SH+ SPACE), egy villogó szorzásjelet látunk (DEF KEY kurzor). A definiált "m" billentyűt lenyomva a gép ki is írja: "31315 bytes free".

Az FN m vagyis MEM függvény Beta nélkül is használható, mivel ROM rutinra támaszkodik. Bárki odaírhatja programja elejére a következőt:

1 DEF FN m//=65535−USR 7962 . Így programírás közben bármikor megtudhatjuk, mennyi helyünk van még PRINT FN m//./

A billentyűkhöz rendelt definíciók a RAMTOP fölé kerülnek, így NEW ellen védettek. Szalagra mentésük a Beta kitöltő rutinjával lehetséges, vagy simán is, ha megkeressük az új RAMTOP-ot (23730-as rendszerváltozó).

LITAUSZKY GYÖRGY



1. ábra

Előző számunkban a PTA—4000-rel készíthető Mandelbrot-képek programjait mutattuk be. Ezúttal a Commodore-hívőknek kedvezünk.

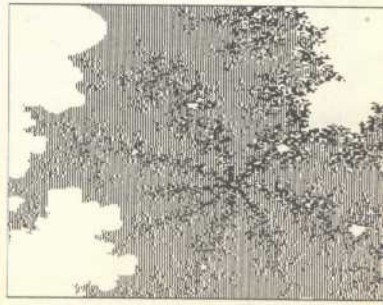
A program alkalmas a Mandelbrot-kép megalkotására, a kép tárolására lemezen, utólagos behívására és átszínezésére is. Lehetőség van arra is, hogy a kép készítését felfüggesszük, majd máskor folytassuk. A kép készítéséhez szükséges adatok: a kiválasztott számhalmaz szélső értékei (valós és képzetes határok), valamint a különböző színekhez rendelt iterációs lépésszám, röviden színhatárok. Ezeket a kép készítése előtt kell megadni; a képelőállítás felfüggesztésekor az adatok szekvenciális fájlban tárolódnak, s a folytatáskor a program behívja őket.

A programot szubrutin blokkokból építettem fel. A 20-69-es sorok között van a menü. A kép előállításához szükséges adatok bekérése a 100-160-as sorok között történik. A vizsgált terület képzetes és valós szám határait a cv( ) valós, illetvea ck( ) képzetes tömbben tárolom, 0 index esetében a kezdeti érték, 1 indexnél a pillanatnyi érték, 2 indexnél a végérték szerepel.

A 200-280-as sorban beállítjuk a nagyfelbontású grafikát (a képernyőt $6000-re), töröljük a képernyő, illetve a színmemória területét, majd a 300-as szubrutin egy vonalkeretet rajzol. Ezután 350-től indul a két főciklus, ahol x és y a nagyfelbontású képernyő pontjainak koordinátái is.

A 880-as szubrutin végzi magát az iterá-

2. ábra



Mandelbrot-program

ciós műveletet, majd a színbeállítás következik, a bitmintázat alapján. A 660-as szubrutin állítja be a megfelelő bitet a színes grafikus képernyőn. A kép előállítása után a program gondoskodik a kép tárolásáról, és a menüre tér vissza.

A tárolt és újra behívott képeket át is színezhetjük, erre a 830-867-es sorok között van mód. 1500-tól 1670-ig terjed a félbehagyott kép folytatásához szükséges adatok beolvasása, majd a kép továbbrajzolása a főciklusnál folytatódik.

Ha a rajzolást abbahagyjuk, az 1700-as sortól elmentődik a félkész kép, illetve fontosabb adatai. Az utóbbiak mindig az „abbahagy" nevű szekvenciális fájlban találhatók, és mivel ez minden újabb abbahagyáskor felülíródik, egy lemezoldalon egyszerre csak egy képen dolgozzunk, hogy ne legyen a folytatáskor keveredés.

3. ábra

A 2000-es szubrutin állítja be a nagyfelbontású színes üzemmódot, a 3000-es törli a grafikus területet. A 3030-as szubrutin rajzolja a keretet, a 4000-es szubrutin állítja vissza a normál képernyőt. Az 5000, 5010, 5020-as szubrutinok egy-egy színt állítanak be a 6500-as színbeállító alaprogram szubruntinjaiként. A 8000-es szubrutin menti el a képet.

A BASIC változatot ne is indítsuk el, mert nagyon lassú. Fordítsuk le először egy fordítóprogrammal; én az Austro-speed-et használom. A lefordított program behívás után RUN-nal indítható, ekkor mindjárt a menü jelentkezik:

F-1 (grafikus üzemmód)
F-3 (kép előállítása)
F-5 (kép folytatása)
F-7 (kép behívása)
F-8 (vége)

F-1 (grafikus üzemmód): láthatóvá válik a színes nagyfelbontású képernyőn, ami persze üres állapotban csak „csillagos ég". Ha előzőleg előállítottunk vagy behívtunk egy képet, akkor az láthatóvá válik. Az F-1 billentyű ismételt lenyomásával újra a menüre léphetünk vissza, a többi funkcióbillentyűvel pedig a színeket állítgathatjuk.

F-3 (kép előállítása): először meg kell adni a kép nevét, majd a valós és képzetes számhatárokat. Az így meghatározott négyszög területét osztja fel a gép 160 × 200-as színes ponthalmazra. Egyre közelebb eső határok megadásával fokozhatjuk a nagyítást. Tapasztalataim szerint néhány milliószoros nagyítást könnyedén el lehet érni. A felbontóképesség határán vízszintes vagy függőleges oszlopszerű képződmények tűnnek fel, attól függően, hogy melyik irányban értük el a maximális nagyítást. Érdemes egyébként lépésről lépésre nagyítani, mert könnyen megeshet, hogy egy fekete területre tévedünk.

A területhatárok kijelölése után a színhatárokat kell megadni (c1, c2, c3, c4). Annak alapján dől el a pont színe, hogy hány iterációs lépést kellett végrehajtani ahhoz, hogy a komplex szám értéke 2 fölé nőjön.

iterációs lépésszám < c1 → bitpár: 00 → szín: fekete

iterációs lépésszám < c2 → bitpár: 01 → 1. szín

iterációs lépésszám < c3 → bitpár: 10 → 2. szín

iterációs lépésszám < c4 → bitpár: 11 → 3. szín

Tulajdonképpen a negyedik színhatár (c4) az iteráció felső határa, ezért ez 360-400-nál ne legyen kevesebb.

```
1 REM ************************************
2 REM ***                              ***
3 REM ***      MANDELBROT - 27         ***
4 REM ***                              ***
5 REM ***   DR HORVATH-PAPP IMRE       ***
6 REM ***                              ***
7 REM ************************************
8 :
20 POKE55,0:POKE56,92:CLR
30 POKE53281,0:POKE53280,0:POKE53272,21
35 BASE=6*4096
38 :
40 PRINT"■■■■■■■■■■■■■ F-1 ■■  GRAFIKUS UZEMMOD"
42 PRINT"■■■■■■■■■■■ F-3 ■■  KEP ELOALLITAS"
52 PRINT"■■■■■■■■■■■ F-5 ■■  KEP FOLYTATAS"
53 PRINT"■■■■■■■■■■■ F-7 ■■  KEP BEHIVAS"
56 PRINT"■■■■■■■■■■■ F-8 ■■  VEGE"
58 ALLJ=0
60 GOSUB1100:REM GET
61 IFW$="■"THENPRINT"■":GOSUB2000:GOTO850
62 IFW$="■"THENGOTO100
64 IFW$="■"THENGOTO1500
66 IFW$="■"THENGOTO830
68 IFW$="■"THENEND
69 GOTO60
90 :
100 INPUT"■■■■■■■■■■■■■MI LESZ A KEP NEVE";N$
102 GOSUB6500:REM SZINBEALLITAS
105 PRINT"■■■■■■ VALOS RESZE KEZDETE";
106 INPUT CV(0)
110 PRINT   "■■■ VALOS RESZE VEGE   ";
111 INPUT CV(2)
112 PRINT   "■■■ KEPZ. RESZE KEZDETE";
113 INPUT CK(0)
115 PRINT   "■■■ KEPZ. RESZE VEGE   ";
116 INPUT CK(2)
120 PRINT"■■■■■■■■■■■ F-1 ■ VISSZA"
122 PRINT"■■■■■■■ F-3 ■ TOVABB"
126 GOSUB1100
128 IFW$="■" THEN GOTO 100
130 IFW$="■" THEN GOTO 132
131 GOTO126
132 X1=0:Y1=1
135 IFCK(0)<CK(2)THENT=CK(0):CK(0)=CK(2):CK(2)=T
138 IFCV(0)>CV(2)THENT=CV(0):CV(0)=CV(2):CV(2)=T
140 :
145 IF(CV(2)-CV(0))=0 THEN GOTO 105
150 GX=(CV(2)-CV(0))/160
152 IF(CK(2)-CK(0))=0 THEN GOTO 105
155 GY=(CK(2)-CK(0))/200
160 INPUT"■■■■■SZINHATAROK:(C1,C2,C3,C4)■■■■■■■■■■■■■■";C1,C2,C3,C4
170 :
200 GOSUB2000:REM HIRES ON
260 GOSUB3000:REM MEMORIA TORLO
280 GOSUB3030:REM KERET RAJZOLO
320 :
350 CK(1)=CK(0)
360 FORY=Y1TO190
380 CV(1)=CV(0)
385 CK(1)=CK(1)+GY
399 :
400 FORX=X1TO316 STEP2
420 CV(1)=CV(1)+GX
440 CX=0:T=0:R=0:S=0:A=0:B=0:Q=0
460 :
480 GOSUB880
500 :
510 IF T>=4 AND CX<C1 THEN S(1)=0:S(2)=0:GOSUB660:GOTO600
511 IF T>=4 AND CX>C3 THEN S(1)=1:S(2)=1:GOSUB660:GOTO600
512 IF T>=4 AND CX>C2 THEN S(1)=1:S(2)=0:GOSUB660:GOTO600
520 IF T>=4 AND CX>C1 THEN S(1)=0:S(2)=1:GOSUB660:GOTO600
540 S(1)=0:S(2)=0:GOSUB660
600 NEXTX,Y:GOTO1700
640 :
660 IF PEEK(203)=57 THEN ALLJ=1
666 IF ALLJ=1 AND X=0 THEN GOTO1700
670 IF S(1)>0THENGOSUB960
680 IF S(2)>0THENX=X+1:GOSUB960:X=X-1
690 RETURN
722 GOSUB8000:REM KEP ELMENTES
780 :
790 GOSUB4000:REM HIRES OFF
800 GOTO 40
824 :
830 INPUT"■■■■■■■■■A KEP NEVE ";N$
832 GOSUB6500:REM SZINBEALLITAS
835 GOSUB2000:REM HIRES ON
836 GOSUB3000:REM MEMORIA TORLO
837 POKE147,0:SYS57812N$,8,1:SYS62631
850 GOSUB1100:REM GET
851 IFW$="■"THENGOSUB4000:GOTO40
852 IFW$="■"THENA1=A1+16:IFA1>255 THENA1=0
853 IFW$="■"THENA2=A2+1:IFA2>255THENA2=0
854 IFW$="■"THENA3=A3+1:IFA3>255 THENA3=0
862 ON(ASC(W$)-133) GOSUB5000,5010,5020
867 GOTO850
875 :
880 IFT>=4 THEN RETURN
890 IFCX>C4 THEN RETURN
900 A=R-S+CV(1):B=2*Q+CK(1)
920 Q=A*B:R=A*A:S=B*B:T=R+S:CX=CX+1
930 GOTO880
952 :
960 CH=INT(X/8):RO=INT(Y/8)
980 LN=Y AND 7
1000 BY=BASE+RO*320+8*CH+LN
1020 BI=7-(XAND7)
1040 POKEBY,PEEK(BY)OR(2↑BI)
1060 RETURN
1090 :
1100 W$="":POKE198,0
1110 GETW$:IFW$=""THENGOTO1110
1120 RETURN
1200 :
1500 INPUT"■■■■■■■■■■■A KEP NEVE:";N$
1502 GOSUB6500:REM SZINBEALLITAS
1504 OPEN3,8,3,"ABBAHAGY,S,R"
1505 INPUT#3,X1
1510 INPUT#3,Y1
1515 INPUT#3,GX
1520 INPUT#3,GY
1525 INPUT#3,CV(0)
1530 INPUT#3,CK(0)
1540 INPUT#3,CV(1)
1545 INPUT#3,CK(1)
1550 INPUT#3,C1
1560 INPUT#3,C2
1570 INPUT#3,C3
1580 INPUT#3,C4
1590 CLOSE3,8,3
1655 GOSUB2000:REM HIRES ON
1658 GOSUB3000:REM MEMORIATORLO
1660 POKE147,0:SYS57812N$,8,1:SYS62631
1670 GOTO360
1680 :
1700 N$="@:"+N$
1706 GOSUB8000:REM KEP ELMENTES
1708 :
1710 OPEN3,8,3,"@:ABBAHAGY,S,W"
1720 PRINT#3,X
1730 PRINT#3,Y
1740 PRINT#3,GX
1750 PRINT#3,GY
1752 PRINT#3,CV(0)
1754 PRINT#3,CK(0)
1760 PRINT#3,CV(1)
1770 PRINT#3,CK(1)
1780 PRINT#3,C1
1790 PRINT#3,C2
1800 PRINT#3,C3
1805 PRINT#3,C4
```

VC20 programok beolvasása

A VC20 számítógép óraüteme nagyobb a C64 üteméné, ezért a magnetofonszalagra eltérő frekvenciával rögzítik az adatokat. Tapasztalataim szerint a VC20-szal be lehet olvasni a C64 programokat, de fordítva nem. Az *1. listán* közölt gépi kódú program átmásolja az interpretert és az operációs rendszert a RAM-ba, majd módosítja a szalagolvasás és a szalagra írás időállandóit. A SYS 52000 parancs kiadása után a C64 be tudja olvasni a VC20 által szalagra mentett programokat vagy adatokat. A POKE 1,53 parancs után újra a saját operációs rendszerével dolgozik a számítógép.

Az átmásolást a *2. lista* BASIC programja több mint egy perc alatt végzi el.

NÉMETH BÉLA

```
C827 A9 E0      150         LDA #$E0
C829 A2 80      160         LDX #80
C82B 20 4A CB   170         JSR SUB         ;OPERACIOS RENDSZER
C82E A2 00      180         LDX #11         ;IDOALLANDOK
C830 A0 16      190         LDY #22         ;CSOKKENTESE
C832 B9 60 CB   200 READ    LDA CIM,Y
C835 8D 42 CB   210         STA POKE+1
C838 B9 61 CB   220         LDA CIM+1,Y
C83B 8D 43 CB   230         STA POKE+2
C83E BD 7D CB   240         LDA ERTEK,X
C841 8D FF FF   250 POKE    STA $FFFF
C844 98         260         DEY
C845 88         270         DEY
C846 CA         280         DEX
C847 10 E9      290         BPL READ
C849 60         300         RTS
C84A 85 FC      310 SUB     STA MUTATO+1
C84C A0 00      320         LDY #0
C84E 84 FB      330         STY MUTATO
C850 B1 FB      340 MASOL   LDA (MUTATO),Y
C852 91 FB      350         STA (MUTATO),Y
C854 E6 FB      360         INC MUTATO
C856 D0 F8      370         BNE MASOL
C858 E6 FC      380         INC MUTATO+1
C85A 8A         390         TXA
C85B C5 FC      400         CMP MUTATO+1
C85D D0 F1      410         BNE MASOL
C85F 60         420         RTS
C860 01 00 D6   430 CIM     .WORD$0001,$FDD6,$F95D,$F979,$F981
C86C 98 F9 AA   440         .WORD$F998,$FBAA,$FBAE,$FB02,$FB04,$FC6B
C878 35 E5 33   450 ERTEK   .BYTE53,$E5,51,41,32,37
C87E 18 51 96   460         .BYTE16,81,150,231,0,182
C884           470 VEGCIM :END

ZEILEN:38   SYMBOLE:8   FEHLER:0

CIM   =C860  ERTEK =C878  MASOL =C850   MUTATO=00FB  POKE  =C841  READ  =C832
SUB   =C84A  VEGCIM=C884
```

1. lista

2. lista

```
100 FORK=4800&TO48151:POKEK,PEEK(K):NEXT
110 FORK=57344TO65535:POKEK,PEEK(K):NEXT
120 POKE64982,229:POKE63837,51
130 POKE63856,41 :POKE63865,32
140 POKE63873,37 :POKE63899,16
150 POKE64426,01 :POKE64430,158
160 POKE64466,231:POKE64468,8
170 POKE64619,182:POKE1,53
```

A halmaz pontjai tehát feketék lesznek (c4-en túl is iterálhatók, a komplex szám értéke mégsem megy 2 fölé), a halmaz közelében lévő pontok az általunk megadott színűek lesznek, a halmaztól távol eső pontok pedig megint csak feketék. A színbeállítást a három színes sávon közvetlenül láthatjuk. Az F-1, F-3, F-5 billentyűvel állíthatjuk tovább a megfelelő színt, a beállítást az F-7 billentyűvel fejezzük be.

Az alapadatok beállítása után célszerű a megfelelő memóriaterületek törlése; ez még lefordítva is kitesz 15-20 másodpercet. A keret megrajzolása után elkezdődik a kép rajzolása, ami 18-36 óráig is eltarthat, attól függően, hogy mennyire dominálnak a képen a halmazon belüli pontok, ezek kiszámítása tart ugyanis legtovább. Én éjszakánként futtatom a programot, így egy-két éjszaka elkészül a kép. A képkészítés felfüggesztéséhez lenyomjuk a balra nyíl billentyűt (a bal felső), és lenyomva tartjuk a következő pont megrajzolásáig, vagy ha ez nem látszik, kb. 30 másodpercig. Ezután meg kell várni, amíg befejeződik a sor, és a félkész kép, valamint adatai elmentésének befejezésével feltűnik a menü.

F-5 (folytatás): a kép nevének megadása és a színek beállítása után a program behívja a képet, és folytatja a rajzolását. Ha elkészült a kép, automatikusan elmentődik, majd ismét megjelenik a képernyőn a menü, ezért hosszabb futás esetén se kapcsoljuk ki a lemezegységet!

F-7 (kép behívása): az elkészült és tárolt képek bármikor behívhatók, és az F-3, F-5, F-7 billentyűkkel átszínezhetők.

F-8 (vége): megjelenik a READY és a kurzor. Ekkor még RUN-nal újraindíthatjuk a programot.

DR. HORVÁTH-PAPP IMRE

Három képet közlünk mutatóban, természetesen csak a képek fekete-fehér változatait. Adataik a következők:

```
1006 CLOSE3,8,3
1010 GOSUB4000:GOTO40
1900 '
2000 M1=PEEK(56576):POKE56576,150:M2=PEEK(648):POKE648,92
2005 M3=PEEK(53272):POKE53272,121
2010 M4=PEEK(53265):POKE53265,PEEK(53265)OR32
2020 M5=PEEK(53270):POKE53270,PEEK(53270)OR16
2030 RETURN
2040 '
3000 '
3005 FOR I=BASE TO BASE+7999 :POKEI,0:NEXTI
3010 FORI=23552 TO 24551:POKEI,A1+A2:NEXTI
3020 FORI=55296 TO 55296+999:POKEI,A3:NEXTI
3025 RETURN
3030 '
3035 V=0:FORX=0TO316:GOSUB960:NEXTX
3040 V=199:FORX=0TO316:GOSUB960:NEXTX
3050 X=1:FORY=0TO200:GOSUB960:NEXTY
3060 X=317:FORY=0TO200:GOSUB960:NEXTY
3070 RETURN
3080 '
4000 POKE53265,M4:POKE53270,M5
4010 POKE56576,M1:POKE53272,M3
4020 POKE648,M2
4030 RETURN
4040 '
5000 FORI=23552 TO 24551:POKEI,(PEEK(I)AND15)ORA1:NEXTI:RETURN
5005 '
```

```
5010 FORI=23552 TO 24551:POKEI,(PEEK(I)AND240)ORA2:NEXTI:RETURN
5015 '
5020 FORI=55296 TO 55296+999:POKEI,A3:NEXTI:RETURN
5025 '
6500 PRINT"J"
6505 K1=4:K2=4:X$="                                   # F-1 =:GOSUB7000:PRINTX$
6510 K1=4:K2=6:X$="                                   # F-3 =:GOSUB7000:PRINTX$
6520 K1=4:K2=8:X$="                                   # F-5 =:GOSUB7000:PRINTX$
6525 K1=4:K2=10:X$="■■   KILEPES          ■  # F-7 =:GOSUB7000:PRINTX$
6530 GOSUB1100
6540 IFA$="="THENK1=4:K2=4:A1=A1+16:IFA1>255THENA1=0
6550 IFA$="="THENK1=4:K2=6:A2=A2+16:IFA2>15THENA2=0
6560 IFA$="="THENK1=4:K2=8:A3=A3+1:IFA3>15THENA3=0
6562 IFA$="="THENRETURN
6565 IFA$="="THENX=INT(A1/16)
6566 IFA$="="THENY=A2
6567 IFA$="="THENZ=A3
6580 FORX=55296+K1+40*K2TO55296+K1+40*K2+16:POKEX,A:NEXT
6590 GOTO6530
6600 '
7000 POKE211,K1:POKE214,K2:SYS58640:RETURN
7010 '
8000 SYS57012*8.0.1
8010 POKE193,0:POKE194,96:REM $6000
8020 POKE174,0:POKE175,128:REM $8000
8030 SYS62957:REM SAVE
8040 RETURN
```

Kép-12: valós határai : -0.75111567 és -0.751114522

képzetes határai : 0.0294705627 és 0.0294705627

színhatárok : 24,130,240,360

Kép-18: valós határai : -0.192593 és -0.185733882

képzetes határai : -0.8177 és -0.82418

színhatárok : 40,100,200,400

Kép-25: valós határai : -1.75947 és -1.75888

képzetes határai : 0.019389 és 0.019058

színhatárok : 37,44,76,360

Floppy monitor

A program egy BASIC nyelven írt egyszerű floppy monitor, amely alkalmas a lemezegység inicializálására, a memória törlésére és kitöltésére, blokk beolvasására a nullás pufferbe, a nullás puffer tartalmának lemezre vitelére, a memória olvasására és írására, valamint 1541-es floppy esetén munkakódok kiadására. Kis módosítással más lemezegység vizsgálatához is jól használható. A karakter—hexadecimális szám és a hexadecimális szám—karakter átalakítások miatt a memóriatartalom kijelzése elég lassú, de alkalmas BASIC fordítóval lefordítva a programot, ez a probléma megoldható.

A program használata

Begépelés után RUN paranccsal indítva, a képernyőn megjelenik egy kétsoros emlékeztető, aminek alapján egyszerűsödik a program kezelése. Az adatokat hexadecimális formában kell megadni. Bármely parancs a téves kiadást követően megszakítható a Q billentyű lenyomásával.

Az utasítások

F1 (I) Inicializálás.

F2 (F) Kitöltés. A monitorok jól ismert F parancsa. Az F2 funkcióbillentyű lenyomását követően az alsó sorban megjelenik egy inverz F betű és a kurzor. Ide kell begépelni az utasítás hely- és értékmeghatározó részét a következő formában: AAAA BBBB CC. Ez azt jelenti, hogy a floppy írható memóriájának AAAA címétől a BBBB címéig CC hexadecimális értékkel tölti fel. Ha az F2 billentyűt a RETURN követi, akkor a bekapcsolási állapotot kapjuk meg. Célszerű utána inicializálást is végezni.

F3 (U1:) Teljes blokk olvasása a nullás pufferbe. A begépelendő AA BB hexadecimális értékek jelentése: AA — sáv, BB — szektor. A beolvasott bájtok a $0300-as címtől kezdődően helyezkednek el (SFD 1001 esetén $1100), és az F5 (M-R) parancs segítségével olvashatók.

F4 (U2:) Teljes blokk írása a lemezre a nullás pufferből. Az AA BB hexadecimális értékek jelentése az előzőhöz hasonló. Ha F4 után a RETURN billentyűt ütjük le, az utoljára beolvasott blokk helyére íródik vissza a nullás puffer tartalma. A nullás puffer a lemezegység memóriájába $0300-tól $0400-ig helyezkedik el.

F5 (M-R) A floppy memóriájának kijelzése egy AAAA hexadecimális címtől kezdődően. Félblokknyi információ (128 bájt) jelenik meg a képernyőn hexadecimális és karakteres formában. Kezdőcím begépelése nélkül a RETURN billentyű hatására az előző F5 (M-R) parancs címétől (ha ilyen nem volt még, $0000-tól) kezdi el a memória kijelzését a program. Így lehetőség nyílik a képernyőn található adatok változtatás utáni gyors felfrissítésére.

F6 (M-W) A memória tartalmának megváltoztatása. AAAA BB jelentése: az AAAA címre BB hexadecimális érték kerül. RETURN lenyomását követően a parancssorban megjelenik az M-W inverzben és az AAAA+1 hexadecimális érték. Ilyenkor a kurzortól kezdődően már csak a BB értékét kell beírni, így az adatbevitel folyamatos. Kilépés a Q billentyű lenyomásával történik.

F7 (NEXT) A következő félblokknyi információ kijelzése. M-R után használva, ezzel az utasítással lehet lapozni.

F8 (LAST) Az előző félblokknyi információ kijelzése. M-R után használva, ezzel az utasítással lehet visszalapozni.

J (JOB) Munkaparancs a nullás puffernek. AA BB CC hexadecimális értékek jelentése: AA — munkakód, BB — sáv, CC — szektor. Csak 1541-es floppyval használható ebből a programból!

Q (BYE) Kilépés a programból, illetve parancs végrehajtása közben kilépés az aktuális parancsból.

PIVARNYIK ATTILA

```
5 PRINT"■";POKE53280,13:POKE53281,12
10 SZ$="          ":X%=0:FORI=1TO9:READT$(I):NEXTI
15 RESTORE:FORI=1TO3:READT1(I):NEXTI
20 FORI=1TO4:READB$(I):NEXTI
25 OPEN1,8,15
30 PRINT"■"SZ$"* (C) FLOPPY-MONITOR ":
35 PRINT"1.0 ■■":PRINT"   ■ F1 ■ I ":
40 PRINT"■ F3 ■ U1  ■ F5 ■ M-R ■ F7 ■";
45 PRINT" NEXT■":PRINT"   ■ F2 ■ F ":
50 PRINT"■ F4 ■ U2  ■ F6 ■ M-W ■ F8 ■";
55 PRINT" LAST":PRINT
60 GOSUB775:FI=RSC(8):FI=FI-132
65 IFFI>4ANDFI<29THEN75
70 IFFI>8ORFI<24THEN60
75 IFFI=8ITHENFI=10
80 IFFI=24THENFI=9
85 ONFI GOTO95,190,285,460,105,225,380
90 FI=FI-7:ONFIGOTO470,480,535
95 PRINT#1,"I"
100 GOSUB625:GOTO60
105 GOSUB700:PRINTSZ$"] ■F■ _■";
110 GOSUB645:IFC=81THENGOSUB735:GOTO60
115 IFLS<>12ANDLS<>0THEN105
120 GOSUB780:PRINTSZ$"] ■WAIT...■";
125 IFLS<>0THEN145
130 PRINT#1,"U";"FI=VAL(T1$)
135 IFVAL(T1$)=FIC2THEN135
140 CLOSE1:OPEN1,8,15:GOSUB625:GOTO60
145 H$=LEFT$(S$,2):GOSUB605:K1=A*256
150 H$=MID$(S$,3,2):GOSUB605:K1=K1+A
155 H$=MID$(S$,6,2):GOSUB605:K2=A*256
160 H$=MID$(S$,9,2):GOSUB605:K2=K2+A
165 H$=MID$(S$,11,2):GOSUB605
170 FIR1=K1+10K2:GOSUB745
175 E$=CHR$(12)+CHR$(T1)+CHR$(1)+CHR$(A)
180 PRINT#1,"M-W"E$
185 NEXT1:GOSUB625:GOTO60
190 GOSUB700:PRINTSZ$"] ■J1 ■ _■";
195 GOSUB645:IFC=81THENGOSUB735:GOTO60
200 IFLS<>5THEN190
205 H$=LEFT$(S$,2):GOSUB605:T=A
210 H$=MID$(S$,4,2):GOSUB605:S=A
215 GOSUB750:IFC=81THENGOSUB735:GOTO60
220 PRINT#1,"I":OPEN2,8,2,"■0"
225 PRINT#1,"U1 ";2;0;T;S:CLOSE2
230 GOSUB625:GOTO60
235 GOSUB700:PRINTSZ$"] ■J2 ■ _■";
240 GOSUB645:IFC=81THENGOSUB735:GOTO60
245 IFLS<>5ANDLS<>0THEN235
250 IFLS=0THEN270
255 H$=LEFT$(S$,2):GOSUB605:T=A
260 H$=MID$(S$,4,2):GOSUB605:S=A
265 GOSUB750:IFC=81THENGOSUB735:GOTO60
270 OPEN2,8,2,"■0"
275 PRINT#1,"U2 ";2;0;T;S:CLOSE2
280 GOSUB625:GOTO60
285 K1=K4:K2=K5:GOSUB700
290 PRINTSZ$"] ■M-R■ _■";
295 GOSUB645:IFLS=81THENGOSUB725:GOTO60
300 IFLS=0THEN320
305 IFLS<>4THEN295
310 H$=LEFT$(S$,2):GOSUB605:K1=A:K4=A
315 H$=MID$(S$,5,2):GOSUB605:K2=A:K5=A
320 PRINT"■■■■■■■■":FORI=1TO16:FORJ=1TO5
325 PRINTS$;:NEXTJ:PRINT"   ";:NEXTI
330 PRINT"■■■■■■■■":FORI=1TO16
335 A=I:GOSUB575:S1$=" "+H$+A=K2
340 GOSUB575:S1$=S1$+H$+A=" ":S2$=""
345 FORJ=1TO2
350 PRINT#1,"M-R"CHR$(K2)CHR$(K1)
355 GET#1,A$:IFA$=""THENA=0:GOTO360
360 A=ASC(A$):GOSUB545:S1$=S1$+H$+A$+" "
365 S2$=S2$+C$:K2=1:GOSUB710:NEXTJ
370 PRINTS1$"■"S2$"■":NEXTI
375 GOSUB635:GOTO60
380 K1=K4:K2=K5:GOSUB700
385 PRINTSZ$"] ■M-W■ _■";
390 GOSUB645:IFC=81THENGOSUB735:GOTO60
395 IFLS=2THEN420
400 IFLS<>7THEN390
405 H$=LEFT$(S$,2):GOSUB605:K1=A:K6=A
410 H$=MID$(S$,7,2):GOSUB605:K2=A:K7=A
415 H$=MID$(S$,6,2):GOSUB605:GOTO425
420 H$=LEFT$(S$,2):GOSUB605
425 E$=CHR$(K2)+CHR$(K1)+CHR$(1)+CHR$(A)
430 PRINT#1,"M-W"E$:K3=1:GOSUB710
435 E$=K1:K7=K2
440 A=K1:GOSUB575:S1$=H$:A=K2:GOSUB575
445 S1$=S1$+H$+SZ$+"■■■■■■■■"
450 GOSUB640:PRINTSZ$"] ■M-W■ "S1$"_■"
455 GOTO390
460 GOSUB700:K1=K4:K2=K5:K3=128
465 GOSUB710:K4=K1:K5=K2:GOTO220
470 GOSUB700:K1=K4:K2=K5:K3=-128
475 GOSUB710:K4=K1:K5=K2:GOTO320
480 GOSUB700:PRINTSZ$"] ■I0■ _■";
485 GOSUB645:IFC=81THENGOSUB735:GOTO60
490 IFLS<>8THEN490
495 H$=MID$(S$,4,2):GOSUB605:K1=A
500 H$=MID$(S$,7,2):GOSUB605:K2=A
505 E$=CHR$(6)+CHR$(0)+CHR$(2)
510 PRINT#1,"M-W"E$CHR$(K1)CHR$(K2)
515 H$=LEFT$(S$,2):GOSUB605
520 E$=CHR$(K)+CHR$(0)+CHR$(1)+CHR$(A)
525 PRINT#1,"M-W"E$
530 GOSUB635:GOTO60
535 CL=SE1:GOTO780
540 PRINTSZ$"] ■BYE...■":SYS64738:END
545 IFA<0THENA=128THENC$=CHR$(A):GOTO565
550 IFNOTA>159ANDA<256>THEN560
555 C$=CHR$(A):GOTO565
560 C$="."
565 E$=CHR$(34)
570 IFC$=E$THENC$=C$+CHR$(20)+E$
575 H$="":I1=INT(A/16):I2=A-I1*16
580 IFI1THENH$=MID$("0123456789ABCDEF",I1+1,1)
585 H$=H$+"0"
590 IFI2THENH$=H$+MID$("0123456789ABCDEF",I2+1,1):GOTO600
595 H$=H$+"0"
600 RETURN
605 H$=LEFT$(H$,1):GOSUB620:I1=A
610 H$=RIGHT$(H$,1):GOSUB620:I2=A
615 A=I1*16+I2:RETURN
620 IFASC(A$)<58THENA=VAL(A$):RETURN
625 A=CHR$(PEEK(SA)-17):SA=VAL(A$)+10
630 RETURN
635 GOSUB700:PRINTSZ$"] ■K...■";:RETURN
640 POKE782,2:POKE781,24:SYS65520:RETURN
645 S$="":
650 GOSUB775:C=RSC(A$)
655 IFNOTC<>91ORC=13>THEN670
660 PRINTCHR$(29)CHR$(20)CHR$(22);
665 LS=LEN(S$):RETURN
670 IFNOTC<>20ORLS>1>""">THEN655
675 PRINTA$;:S$=LEFT$(S$,LEN(S$)-1)
680 GOTO650
685 IFA2CCRNDC250THEN655
690 IFNOTC<>CC71ANDC264ORC=32>THEN650
695 PRINTA$"_■";:S$=S$+A$:GOTO650
700 GOSUB640:FORI=1TO4:PRINTSZ$:NEXTI
705 GOSUB640:RETURN
710 K2=K2+K3:IFK2<0THENK2=K2+256:K1=K1-1
715 IFK1<0THENK1=0:K2=0:RETURN
720 IFK2>255THENK2=K2-256:K1=K1+1
725 IFK1>255THENK1=0:K2=0
730 RETURN
735 GOSUB700:PRINTSZ$"] ■ERR...■";
740 RETURN
745 I1=INT(I/256):I2=I-I1*256:RETURN
750 IFT<10RT>350RS<0THENC=91:RETURN
755 FORI=1TO4
760 IFNOTCT(T1(I)ANDT>T1(I+4))THEN720
765 IFS>S1(I)THENC=91
770 NEXTI:RETURN
775 POKE198,0:WAIT198,1:GET#1,A$:RETURN
780 DATA0,17,24,30,19,25,31,26,20,19
785 DATA17,16
```

PTA—4000

Disassembler program

Bizonyára hasznos segítőtársa lesz a PTA—4000 (PC—1500) használóinak az alábbi disassembler program. A szoftverhiány késztetett arra, hogy hozzáfogjak a megírásához. Segítségével számos hasznos

1. lista

```
                &88C8-&83C8

88 63 68 23 88 43 A8 84 48 8D 68 2D
88 8C 88 2C 88 13 88 93 61 82 81 62
81 42 81 67 81 47 61 28 A2 88 81 75
81 55 81 79 81 59 82 C3 62 C8 42 2F
62 2F 43 22 83 62 43 64 CE FF FF 51
64 28 84 4E 87 88 48 6D 48 33 89 8C
4B 23 6B 23 4B A3 AB 83 6B C8 8B 8E
8D 8E 4E C5 4E 31 6E 31 8F 8E 6F 12
91 83 51 64 51 AB 71 8E 91 4E 91 AF
91 CF 71 33 71 D3 52 61 72 21 52 A3
92 83 52 64 72 87 92 47 52 A8 72 C8
92 89 92 AF 92 CF 52 33 82 13 73 C8
53 33 95 62 95 42 75 44 75 4C 85 75
95 79 95 59 95 47 95 67 28 28 28 28
28 61 62 29 28 28 28 28 28 28 28 28
41 28 28 28 50 28 28 28 55 4C 28 28
28 28 29 28 28 28 4C 28 28 28 48 28
53 28 28 28 78 79 7A 7B 7C 7D 7E 7F
80 FF 79 7B 7D 7F FD FF 82 88 88 B3
81 14 82 88 89 F1 83 88 89 85 14 8E
8A 12 9E 8A 13 83 8B 12 93 8B 13 81
8C 12 91 8C 13 87 8D 12 97 8D 13 85
8E 12 95 8E 13 8F 8F 14 8F 11 12 9F
11 13 8D 12 12 9D 12 13 8B 13 12 9B
13 13 89 14 12 99 14 13 F7 16 88 86
97 88 86 97 89 B7 18 14 4E 98 18 4C
98 19 DF 1B 84 42 9B 88 46 9B 88 BD
1E 18 DD 21 84 48 A1 88 44 A1 88 BA
23 20 84 A4 88 84 A4 88 B5 25 14 4A
A5 18 48 A5 19 AA 25 2A 47 A6 88 45
A8 88 88 29 16 38 2A 88 88 2D 14 F9
31 88 D1 34 88 DB 33 88 E3 35 88 88
36 88 8A 37 88 9A 38 88 88 B9 88 88
B9 89 B1 3A 18 43 3B 88 FB 3D 88 D9
3E 88 D5 3F 88 41 C1 88 BE 42 28 E1
43 88 A8 44 88 8A C5 88 88 C5 89 F5
47 88 C3 49 18 C1 4A 18 CD 4C 18 CF
4D 18 CB 4E 18 C9 4F 18 C5 50 18 C7
51 18 C8 4B 88 D8 4B 28 E8 4B 88 F8
4B 88 C2 4B 38 D2 4B 28 E2 4B 18 F2
4B 88 C4 4B 38 D4 4B 18 E4 4B 18 F4
4B 28 C6 4B 88 D6 4B 18 E6 4B 88 F6
4B 28 C8 4B 18 D8 4B 88 E8 4B 18 F8
4B 88 CA 4B 18 DA 4B 88 EA 4B 88 FA
4B 88 CC 4B 18 DC 4B 88 EC 4B 88 FC
4B 88 CE 4B 38 DE 4B 18 EE 4B 88 FE
4B 88 83 88 87 A3 88 21 4F 81 17 EF
81 31 89 84 87 A9 84 21 49 85 17 E9
85 31 4D 8F 17 ED 8F 31 8F 98 87 AF
18 21 87 97 87 A7 17 21 8C 99 87 8C
9A 87 D7 1C 87 D3 1D 87 8D 9F 87 AD
1F 21 85 A4 87 A5 24 21 88 AC 87 AB
2C 21 4B AD 17 EB 2D 31 81 B9 87 A1
39 21 8E C5 87 AE 45 21 CA 82 88 CE
86 88 DE 87 88 CC 88 88 EC 89 88 8E
15 88 42 9B 89 B1 28 88 48 A1 89 BA
22 88 88 A7 88 48 27 8A 58 27 85 4C
2B 88 8A 2E 84 8A AE 88 C8 2F 84 88
AF 88 C8 38 88 8E 32 88 C1 3C 88 81
48 88 4A C6 88 4E 46 8A 5E 46 85 AA
48 88 68 78 6A C8 84 8E 83 B4 61 84
8E 83 B5 61 84 8E 83 B4 61 84 8E 83
B5 61 B5 88 2E 9A F1 B9 8F B7 8A 83
83 B3 38 9A B3 36 9A
```

```
'0:"A"CLEAR :CSIZE 1:COLOR 1:INPUT
  "KEZDOCIM=";X
20:S=0:A=X:GOSUB 240:LCURSOR 0:
  LPRINT A0;
30:B=PEEK X: IF B=253GOTO 60
40:H=58:FOR I=422TO 839STEP 3:GOTO
  250
50:NEXT I:GOTO 90
60:X=X+1:B=PEEK X:LCURSOR 6:LPRINT
  "FD      B";
70:H=88:FOR I=752TO 917STEP 3:GOTO
  250
80:NEXT I
90:LCURSOR 13:LPRINT "?";:FOR I=406
  TO 419:IF PEEK I=B6GOTO 220
100:NEXT I:FOR I=420TO 421:IF PEEK I
  =B6GOTO 210
110:NEXT I:GOTO 230
120:LCURSOR 18:LPRINT B0;
130:LCURSOR 8:A=B:GOSUB 240:LPRINT C
  0;
140:E=((PEEK (I+1))AND 127)%2+198:F=
  ((PEEK E)AND 31)+65:G=(PEEK (E+1
  )AND 31)+65
150:H=((PEEK (E))AND 224)/4+((PEEK (
  E+1))AND 224)/32+48
160:LCURSOR 12:LPRINT CHR$ F;CHR$ G;
  CHR$ H;
170:E=(PEEK (I+2)AND 15)%4+362:
  LCURSOR 17
180:LPRINT CHR$ PEEK E;CHR$ PEEK (E+
  1);CHR$ PEEK (E+2);CHR$ PEEK (E+
  3);
190:E=(PEEK (I+2)AND 48)/16+1:ON E
  GOTO 230,220,210,200
200:A=PEEK (X+3):GOSUB 240:LCURSOR 2
  7:LPRINT C0;:S=S+1
210:A=PEEK (X+2):GOSUB 240:LCURSOR 2
  5:LPRINT C0;:S=S+1
220:A=PEEK (X+1):GOSUB 240:LCURSOR 2
  3:LPRINT C0;:S=S+1
230:LF 1:X=X+S+1:GOTO 20
240:U=INT (A/256):C=A-U%256:POKE &79
  04,%B2,U,C:CALL 920,A:C$=MID$ (A
  $,3,2):RETURN
250:D=(PEEK (I+1))AND 128:IF D=128
  GOTO 280
260:IF PEEK I=B6GOTO 130
270:GOTO H
280:IF PEEK I=BTHEN LET B$="X":GOTO
  120
290:IF (PEEK I)+16=BTHEN LET B$="Y":
  GOTO 120
300:IF (PEEK I)+32=BTHEN LET B$="U":
  GOTO 120
310:GOTO 270
```

2. lista

és praktikus ROM-rutin visszafejthető, s így jelentősen növekszik a gépen megoldható feladatok száma.

A program egy adatmezőből (*1. lista*) és az ezt kezelő BASIC programból (*2. lista*) áll.

A BASIC program felépítése

10—20 Kezdőértékek beállítása, az első cím kinyomtatása.

30 Megvizsgálja, hogy a kérdéses utasítás az 1. vagy a 2. lapra vonatkozik-e.

40—50 A 1. lapra vonatkozó utasítások közül a kérdéses kikeresése az adatmezőből.

60—80 A 2. lapra vonatkozó utasítások közül a kérdéses kikeresése az adatmezőből.

90—110 Megállapítja az ismeretlen utasításokhoz tartozó adatbájtok számát.

120—180 Az utasításkód és a mnemonik kinyomtatása.

190—200 Az adatbájtok számának megállapítása és kinyomtatásuk.

230 Az adatbájtok utáni első utasításbájt megkeresése, a 20-as sorba visszatérve a folyamat újra indul.

240 Decimális-hexadecimális konverter.

250—310 Sorutasítás-vizsgáló rutin.

A BASIC program felépítése olyan, hogy időigényesebb részei gépi nyelven is könnyen megírhatók, és egyszerű a program assembler kiegészítése is. A program a bővítetlen gépeken is futtatható, ekkor a BASIC program 240-es sorában lévő címet és

```
AB91    BE   SJP          B0EB
AB94    BE   SJP          B116
AB97    FD98 PSH$  Y
AB99    BE   SJP          ACBB
AB9C    B5   LDI  A       02
AB9E    AE   STA  (ab)    79F4
ABA1    BE   SJP          AAD9
ABA4    BE   SJP          A9E4
ABA7    BE   SJP          ABE1
ABAA    BE   SJP          A386
ABAD    6A   LDI  UL      00
ABAF    4A   LDI  XL      00
ABB1    BE   SJP          A20E
ABB4    FDA8 PSH$  U
ABB6    BE   SJP          A519
ABB9    FD98 PSH$  Y
ABBB    B5   LDI  A       19
ABBD    BE   SJP          AC2F
ABC0    B5   LDI  A       19
ABC2    BE   SJP          AC2F
ABC5    FD1A POP$  Y
ABC7    FD2A POP$  U
ABC9    60   INC  UL
ABCA    6E   CPI  UL      04
ABCC    8B   BZS  +       04
ABCE    4A   LDI  XL      20
ABD0    9E   BCH  -       21
ABD2    BE   SJP          A9E4
ABD5    6A   LDI  UL      00
ABD7    BE   SJP          A519
ABDA    BE   SJP          ABE1
ABDD    FD1A POP$  Y
ABDF    8E   BCH  +       D6
ABE1    B5   LDI  A       02
ABE3    AE   STA  (ab)    79F4
ABE6    B5   LDI  A       04
ABE8    43   SDE
ABE9    B5   LDI  A       00
ABEB    8E   STA  (X)
ABEC    BA   JMP          AA04
ABEF    BE   SJP          AC97
ABF2    B5   LDI  A       00
ABF4    AE   STA  (ab)    79E7
ABF7    AE   STA  (ab)    79E8
ABFA    AE   STA  (ab)    79EA
ABFD    AE   STA  (ab)    79E8
AC00    AE   STA  (ab)    79F2
AC03    AE   STA  (ab)    79F8
AC06    9A   RTN
```

3. lista

a &398-as címen kezdődő konvertáló rutin megfelelő abszolút ugróutasításait kell átírni.

A *3. listán* a program által disassemblált TEST rutin látható.

KABAI IMRE

HELP bővítő

Szívesen dolgozom a Print Technik 8 k-s BASIC bővítésével. A program módosításának, illetve kiegészítésének gondolata akkor merült fel, amikor a C64 „ROM alatti" területeire lettem kíváncsi; például a SIMON's BASIC, az Easy Script stb. programok egy része itt, RAM-ban található.

A megírt kiegészítés így elsősorban a ROM-ok alatti RAM-ok vizsgálatát és használatát segíti.

A bővítés alapja egy lemezen meglévő HELP+ program, mely két részből áll, egy

```
400 PRINT"□□□□□□□TURELMET KEREK..."
500 IFA=0THENA=1:LOAD"HELP-1",8,1
510 IFA=1THENA=2:LOAD"HELP-2",8,1
610 POKE52,128:POKE56,128
620 CLR
630
730
760 POKE768,139:POKE769,227:POKE772,124
770 POKE773,165:POKE644,128
780 POKE195,1:POKE196,9
790
810 SYS33000
820 SYS40744
870 REM FLOPPY RAM-BA FORDITAS
880 REM POKE39014,135 -TEL LEHETSEGES
890 NEW
```

1. lista

BASIC-betöltőből és egy gépi kódú programból. Ez utóbbinak végcíme: $9F02, így a bővítő gépi kódú program elhelyezhető a $9F03-$9FFF tárterületen; jelen esetben ez $9F28-$9FFF.

A program további szolgáltatásai

A HELP+ monitor üzemmódjában a program írja-olvassa a $A000-$BFFF, $E000-$FFF és a $D000-$DFFF címtartományú RAM-ot. Ez utóbbi területhez úgy lehet hozzáférni, hogy a programban kikapcsoljuk mind a BASIC, mind a KERNAL ROM-ot, tehát nem azonos azzal a $D000-$DFFF címtartománnyal, melyet többek között a VIC, a SID és a CIA vesz igénybe. A $9FDA rekesz megfelelő átírásával ($32) természetesen olvashatjuk a karakter ROM-ot is.

A HELP+ assembler üzemmódjában lehetőség van arra, hogy assembly programot mint gépi kódú programot fordítsunk be a fent említett területekre ($A000-$BFFF, illetve $D000-$FFFF). A lemezegység puffertárába (RAM) is fordíthatunk be assembly programot (8-as egység). Ez a floppy $300-$7FF tárterületét jelenti; a VC 1541 lemezegységgel próbáltam ki.

Fordítás előtt a POKE39014, 135 parancsot kell kiadni. A fordítás befejeztével a rekesz tartalma felveszi eredeti értékét,

```
9F28  A9 20     LDA  #20
9F2A  A2 9F     LDX  #9F
9F2C  A0 EA     LDY  #EA
9F2E  8D 18 8C  STA  8C18
9F31  8D 98 8C  STA  8C98
9F34  8D B8 8C  STA  8CB8
9F37  8D 65 98  STA  9865
9F3A  8E 23 80  STX  8023
9F3D  8E 1A 8C  STX  8C1A
9F40  8E 9A 8C  STX  8C9A
9F43  8E B2 8C  STX  8CB2
9F46  8E 6E 91  STX  916E
9F49  8E 67 98  STX  9867
9F4C  8C 1B 8C  STY  8C1B
9F4F  8C 9B 8C  STY  8C9B
9F52  8C B3 8C  STY  8CB3
9F55  8C 68 98  STY  9868
9F58  8C 69 98  STY  9869
9F5B  A9 7D     LDA  #7D
9F5D  A2 4C     LDX  #4C
9F5F  A0 B6     LDY  #B6
9F61  8D 22 80  STA  8022
9F64  8E 6C 91  STX  916C
9F67  8C 6D 91  STY  916D
9F6A  A9 D8     LDA  #D8
9F6C  A2 E7     LDX  #E7
9F6E  A0 F3     LDY  #F3
9F70  8D 99 8C  STA  8C99
9F73  8D B1 8C  STA  8CB1
9F76  8E 19 8C  STX  8C19
9F79  8C 66 98  STY  9866
9F7C  60        RTS
9F7D  A9 80     LDA  #80
9F7F  85 38     STA  38
9F81  8D 84 02  STA  0284
9F84  4C 22 E4  JMP  E422
9F87  8A        TXA
9F88  48        PHA
9F89  A5 FB     LDA  FB
9F8B  48        PHA
9F8C  A5 FC     LDA  FC
9F8E  48        PHA
9F8F  A5 FD     LDA  FD
9F91  48        PHA
9F92  A2 57     LDX  #57
9F94  B9 4E 00  LDA  004E,Y
9F97  85 FD     STA  FD
9F99  98        TYA
9F9A  18        CLC
9F9B  65 41     ADC  41
9F9D  85 FB     STA  FB
9F9F  A9 00     LDA  #00
9FA1  65 42     ADC  42
9FA3  85 FC     STA  FC
9FA5  A9 08     LDA  #08
9FA7  20 CC 8C  JSR  8CCC
9FAA  68        PLA
9FAB  85 FD     STA  FD
9FAD  68        PLA
9FAE  85 FC     STA  FC
9FB0  68        PLA
9FB1  85 FB     STA  FB
9FB3  68        PLA
9FB4  AA        TAX
9FB5  60        RTS
9FB6  A9 F3     LDA  #F3
9FB8  8D 66 98  STA  9866
9FBB  AD 88 02  LDA  0288
9FBE  09 80     ORA  #80
9FC0  A8        TAY
9FC1  A9 00     LDA  #00
9FC3  AA        TAX
9FC4  94 D9     STY  D9,X
9FC6  18        CLC
9FC7  69 28     ADC  #28
9FC9  90 01     BCC  9FCC
9FCB  C8        INY
9FCC  E8        INX
9FCD  E0 1A     CPX  #1A
9FCF  D0 F3     BNE  9FC4
9FD1  A9 FF     LDA  #FF
9FD3  95 D9     STA  D9,X
9FD5  4C B6 E3  JMP  E3B6
9FD8  78        SEI
9FD9  A9 34     LDA  #34
9FDB  85 01     STA  01
9FDD  B1 FB     LDA  (FB),Y
9FDF  85 FD     STA  FD
9FE1  A9 37     LDA  #37
9FE3  85 01     STA  01
9FE5  58        CLI
9FE6  60        RTS
9FE7  78        SEI
9FE8  A9 34     LDA  #34
9FEA  85 01     STA  01
9FEC  A5 FD     LDA  FD
```

```
9FEE  91 FB     STA  (FB),Y
9FF0  4C E1 9F  JMP  9FE1
9FF3  78        SEI
9FF4  A9 34     LDA  #34
9FF6  85 01     STA  01
9FF8  B9 4E 00  LDA  004E,Y
9FFB  91 41     STA  (41),Y
9FFD  4C E1 9F  JMP  9FE1
```

2. lista

vagyis a következő fordítás ismét a C64 memóriájába megy.

A lemezegység RAM-ban lévő gépi kódú program futtatható a „M — E" utasítással, a HELP+ monitor üzemmódjában pedig a * jellel.

A lemezes HELP egy SYS 64738 (RESET) után, bejelentkezésének megfelelően, a BASIC RAM végét $A000-ra állítja. Egy futó BASIC program elronthatja a HELP+ végét. Ekkor a sztringterület eleje $A000! Védelem ez ellen: RESET után a BASIC RAM vége $8000-ra állítódik, s a szabad BASIC bájtok száma is ennek megfelelően íródik ki a képernyőre.

A HELP bővítő kialakítása, használata

A már meglévő BASIC-töltő programot módosítani kell az *1. lista* alapján. A *2. lista* szerinti HELP—2 nevű gépi kódú programot lemezre kell rögzíteni. Így a BASIC-töltő futásakor a meglévő HELP—1 gépi program mellett betöltődik és lefut a mi HELP—2 nevű programunk is.

A program használata a bejelentkezés után a szokásos. Ha ismét a BASIC vagy KERNAL ROM területét, illetve a VIC, SID, CIA rekeszeit szeretnék vizsgálni monitor üzemmódban, akkor a $9FDA, $9FE9 regiszterek tartalmát $37-re kell átírnunk.

Megjegyzés: A tár 1-es címének olvasásakor monitor üzemmódban $34-et kapunk, mivel ekkor ki van kapcsolva a KERNAL és BASIC ROM.

BÖDÖR LAJOS

BITEK ÉS FIGURÁK

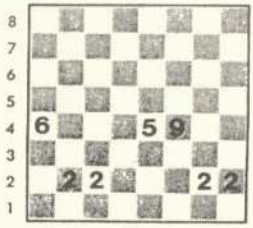
Megnyitáselmélet és pozíciótábla

Az állásértékelésnek két fő tőréspontja van. Az egyik: átmenet a megnyitásból a középjátékba, a másik pedig: átmenet a középjátékból a végjátékba. Nagyon nehéz meghatározni, hogy meddig tartanak az egyes részek, mert meglehetősen szubjektív, hogy a szóban forgó játszmában a lépéseket meddig tekintjük a megnyitáshoz tartozónak, és mikor kezdődik a középjáték. Ugyanígy: hol van a határ a középjáték és a végjáték között? Egyes programok a lépésszámtól teszik ezt függővé, mások pedig a táblán lévő anyagi érték összegétől.

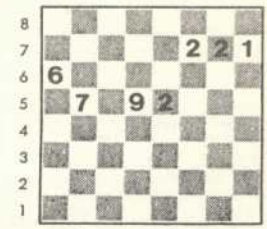
A megnyitás és a középjáték közötti átmenetet általában a lépett figurák számával határozzák meg. Ez egyszerű, de hatásos módszer. Nagy segítséget nyújt az is, hogy a programok általában az első lépéseket nem számítják, hanem egy — adatbázisszerű — megnyitási könyvtárból veszik. Ez a megnyitási könyvtár a különböző programoknál és az eltérő típusú sakkszámítógépeknél más és más; nagyságuk is sokban eltérhet. Így például az ismert CRAY BLITZ program 5000 lépést, a MEPHISTO 20 ezer lépést, az AWIT 10 ezer lépést tartalmaz; a KEMPELEN C64-be e sorok írója 3000 lépést táplált be.

A megnyitási könyvtárat tanácsos mindkét színhez különkülön elkészíteni. Az egyik változat világosnak csak a jó lépéseit tartalmazza, viszont az összes szóba jövő lehetséges sötét válaszlépés benne van; a másik pedig csak sötét jó lépéseit tartalmazza az összes lehetséges világos válaszlépéssel. Ennek az a célja, hogy ha például a gép játszik világosként, akkor azt a könyvtárat veszi, amelyikben világos jó lépései és az összes sötét válaszlépés található. A program a könyvtárból csak jó lépéseket adhat, és megtalálja a rá vonatkozó válaszlépéseket is. Így az algoritmus nem tévedhet el a megnyitási változatok útvesztőjében, viszont követhet egy „gyilkos" könyvtárat, amelynek lényege, hogy olyan változatokba tereli a játékot, amelyek könyvtára nagyon részletesen ki van dolgozva, és tartalmazza az ide vonatkozó megnyitási sakkcsapdákat is.

A megnyitási könyvtárat nagyon precízen kell kidolgozni, mert főként ez határozza meg a későbbiekben a játék stílusát. Nem mindegy, hogy a megnyitás végállásától az algoritmusnak egy éles szicíliai védelmet kell folytatnia, avagy egy vezércselt. Ezért a megnyitás lépéseit úgy a legcélszerűbb kiválasztani, hogy az elkészült sakkprogrammal több játszmát is lejátszunk: kipróbáljuk, hogy egyes helyzetekre hogyan reagál, melyik lépést találja meg magától is az algoritmus, és hol szorul segítségre. Ezek alapján megállapíthatjuk, melyik megnyitási rendszert kezeli jól az algoritmusunk, és ennek alapján döntjük el, hogy mely megnyitásokból állítjuk össze a könyvtárat. Ha nincs elég memóriánk, akkor főként azokkal a lépésekkel bővítsük a könyvtárat, amelyek helyett a program algoritmusa rosszakat lépne. Ezek után bővíthetjük egyéb lépésekkel, hogy a játszma végére több idő maradjon. Ugyanis a megnyitás-

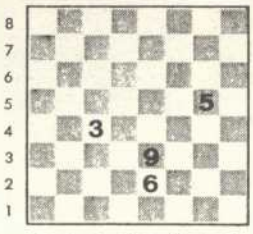


3/a. ábra. Világos gyalog pozíciós táblája

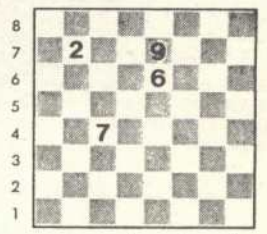


3/b. ábra. Sötét gyalog pozíciós táblája

Az üresen hagyott mezőknek zérus az értékük



4/a. ábra. Világos futó pozíciós táblája



4/b. ábra. Sötét futó pozíciós táblája

Az üresen hagyott mezőknek zérus az értékük

1. ábra. A szicíliai védelem egyik közismert változatának alapállása

2/a. ábra. Világos király pozíciós táblája

2/b. ábra. Sötét király pozíciós táblája

Az üresen hagyott mezőknek zérus az értékük

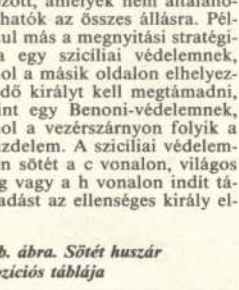
5/a. ábra. Világos huszár pozíciós táblája

sal megtakarított időt a versenyfokozaton a program a közép- és végjátékban felhasználhatja, ezzel is növelve a program játékerejét.

A megnyitási könyvtár a legtöbb sakkprogramnál konkrét lépések sorozatát tartalmazza, aminek az a hátránya, hogy a program nem ismeri fel a lépéscserével létrejövő állásokat. Hacsak azok is nincsenek letárolva a memóriában, az algoritmus nem tudja azonosítani az állást, és kénytelen a saját értékelőfüggvényét használni. Egyes sakkprogramok ezért főként nem konkrét lépéseket tárolnak, hanem minden egyes megnyitásra külön algoritmust írtak, amelyek az egyes megnyitások stratégiai elveit súlyozzák nagyobb mértékben. Előny a kisebb tárkapacitás, hátrány, hogy a program már az első lépésekkel is időt veszít.

A KEMPELEN ATARI program e kettő közötti átmenetet jelenti: a konkrét lépések vannak betáplálva, de amikor elfogynak ezek a lépések, akkor a könyvtár az értékelő algoritmusnak egy súlyozott sakktáblát ad át, amelyet pozíciós táblának neveztünk el. Ennek a táblának a súlyozása a megnyitástól függően változik. A pozíciós táblával a program minősíthet a különféle megnyitásokból kialakult állások között, így a megnyitások más-más pozícióiban különbséget tehet az egyes stratégiák között, amelyek nem általánosíthatók az összes állásra. Például más a megnyitási stratégiája egy sziciliai védelemnek, ahol a másik oldalon elhelyezkedő királyt kell megtámadni, mint egy Benoni-védelemnek, ahol a vezérszárnyon folyik a küzdelem. A sziciliai védelemben sötét a c vonalon, világos a g vagy a h vonalon indít támadást az ellenséges király ellen. A Benoni-megnyitás sötétje a világos gyalogcentrum alániázására, fellazítására törekszik b7—b5 lépéssel, míg világos vagy az e4—e5 áttörést kísérli meg, vagy a b2—b4—b5 stratégiát választja.

5/b. ábra. Sötét huszár pozíciós táblája



Hasonló stratégiai elemek figyelembevétele nagyon egyszerű a pozíciós táblával, amelyet elkészíthetünk külön-külön minden figuratípusra az egyes megnyitási változatoknál, mind világossal, mind sötéttel. Ekkor tehát tizenkét pozíciós táblát kell kezelnünk megnyitásonként, de megéri a fáradságot. A következőt javaslom.

Egy teljesen külön memóriarészben tároljuk ezeket a pozíciós táblákat, olyan módon, hogy tizenkettesével (figuratipusonként), sorszám szerint tudjunk rájuk hivatkozni, így a megnyitási lépéssorozatban elég csak egy-egy számot tárolni — amikor megváltoztatjuk a pozíciós táblákat. Ha elfogynak a könyvtár lépései, akkor az értékelőfüggvény az utoljára megtett lépéssel tárolt pozíciós táblák sorszámát kapja meg. Ezeket a táblákat átmásolja magának a megfelelő helyre, és így dolgozik tovább.

A pozíciós táblák súlyozását többféle módon szervezhetjük. A legegyszerűbb az egyes figurák kedvező elhelyezkedési mezőjét felpontozni. Ezzel kompenzálhatjuk például azt az esetet, amikor a sötét vezér a5-re lép c7 helyett, mivel a5-ön nagyobb lesz a mozgékonysága és a centrumértéke. Az *1. ábrán* bemutatott állás pozíciótábláit láthatjuk a *2—5. ábrákon.*

A megnyitási könyvtár lépéskiválasztásának másik szempontja, hogy olyan lépéseket válasszunk ki, amelyek során a könyvtárból való kilépés pillanatában a gyalogok helyzete igazodik a kibontakozó középjátékhoz. Vagyis törekedjünk arra, hogy a játszó algoritmus kevés gyaloglépést tegyen. Minden egyes gyaloglépést nagyon meg kell fontolni, mert ezzel a bábbal tilos a visszafelé lépés, és az esetleges rossz lépést képtelenség visszavenni. Ha a megnyitásban már kialakítjuk a középjáték végleges gyalogszerkezetét, akkor az értékelőfüggvényben a gyaloglépéseket megfelelően leponthatjuk, kivéve a gyalogroham vagy gyalogtámadás esetét. Így nagyobb az esélye, hogy programunk nem lép gyaloggal rosszul.

KOVÁCS P. ATTILA

Interaktív kompakt lemezek

A képen látható szedéssel, kiemelten hozta a hírt a 68 Micro nevű szaklap arról, hogy a Sony és Philips cégek szabványosították az interaktív kompakt lemezeket, és az ezeket kezelő szoftvert a Microware nevű cég, és nem a Digital Research (a CP/M készítője), nem a Microsoft (az MSDOS készítője) írta.

Miért olyan fontos ez az esemény? A kompakt lemezek kapacitása sokkal nagyobb, mint a hagyományos hanglemezeké, és lényegesen jobb a hangminőségük. Ezen okok miatt a hanglemezgyártás újra virágzásnak indult, sőt: az eladott darabok száma új csúcsokat ért el. Természetesnek tűnt, hogy digitális információtárolásuk miatt érvényesülnek a számítástechnikában is. Az ún. interaktív változatokra ez máris igaz.

Mi az interaktív lemez? Egy speciális lejátszón leforgatható kompakt lemez. Mint fontos tulajdonság megemlítendő, hogy az ezekhez készült lejátszók elfogadják a közönséges kompakt lemezeket is, és néhány interaktív lemez hagyományos lejátszón is használható. (Ez utóbbi esetben azonban az új tulajdonságok nem érvényesülnek.) Az új és számunkra alapvető fontosságú tény: kódolt képek és adatok tárolhatók!

Az interaktív lemezeknek a mágneslemezekhez hasonlóan van tartalomjegyzék és állomány része. A fájlleíró rész a fájl típusára vonatkozó információt is tartalmaz (a fájl lehet: hang, kép vagy egyéb típusú). Az egyes fájlokon belül a típus blokkonként változhat. A lemez kezelését a lejátszóban tárolt operációs rendszer végzi. Ez az új szabvány szerint az OS9 elnevezésű. (Erről az operációs rendszerről a következő számban adunk ismertetést. Itt csak annyit: ez egy több felhasználós, osztott idejű rendszer.) Az operációs rendszer a fájl típusától függően két videochipen, egy audiochipen keresztül, vagy közvetlenül olvassa az információkat. (A hardver egyébként egy Motorola 68000 típusú — tehát 16 bites — mikroprocesszorú célszámítógép.)

A hangok tárolását az eddig megszokottaktól eltérően a jel deriváltjával oldják meg. A lemez kapacitása 640 Mbájt. A keresési idő igen kedvező, de az átviteli sebesség nem éri el a merev lemezeknél megszokottat.

Ezek miatt a kezelő szoftver igyekszik az információkat a lehető legkedvezőbb helyzetben tartani, ezért a fájlnyitáskor egyúttal megkeresi az elérési idő szempontjából még rendelkezésre álló legjobb helyet. Az OS9 rendszer hagyományos változata a fájlleíró részt a fájl olvasása közben kapja meg, ezért hangtípusú információk esetén nem használható. Az új változatnál a fájl elé teszik a leíró részt. A fájlok itt mindig statikusak!

A szabvány alapvetően a hang rögzítésére szolgáló rendszerekre készült, de szövegek, adatbázisok, nagyméretű programok tárolását is célul tűzték ki. A nem elégséges sávszélesség miatt a képek minősége jelenleg még nem olyan jó, mint a videoszalagoknál.

Mivel a piacot uraló két cég ezt az operációs rendszert fogadta el, ebből várhatóan többet fognak értékesíteni, mint az eddig bármilyen célra kidolgozott operációs rendszerekből.

Kétórás játékok – miért ne?

COMMODORE 64

A Commodore 64-re eddig viszonylag kevés számú játékóra-programot publikáltak. Ezek közös jellemzője, hogy csak *egy* óra működését tették lehetővé.

Egy részük BASIC nyelven íródott, és az elindított órát csak RUN/STOP-pal lehetett megállítani. Másik részük már lényegesen jobb volt, mivel assembler nyelven készültek: az órát programból vagy valamelyik billentyű lenyomásával lehetett elindítani, és ugyanilyen módon kellett megállítani. Az igényesebb felhasználást azonban akadályozta, hogy a leállítás, majd az újbóli indítás után az óra nem a képernyőn kijelzett időről indult.

Eredetileg olyan programot akartam készíteni, amely megfelel az alábbi követelményeknek:

— assembly nyelven megírva két óra váltakozva működjön;

— az órák a képernyőn kijelzett (leállított) időértékről induljanak; programból és billentyűzetről egyaránt működtethetőek legyenek;

— a két óra közül bármelyik indítható legyen elsőként,

— újbóli betöltés nélkül az órák ismét aktivizálhatók és hívhatók legyenek.

A Mikroszámítógép Magazin 1985/6. számában Müller Gábor—Laky Zsolt „Megszakítás I." című cikkükben a szerzők bemutatják, hogy a CIA-chipek megszakítási rendszerét miként használhatjuk fel saját céljainkra. Bevezetik az ENGED bájt fogalmát, melynek programozásával az óra indítása-leállítása vezérelhető.

Azzal, hogy az előbb leállított óra az újraindításkor milyen időértékről indul, a szerzők nem foglalkoztak, mivel céljuk az ébresztés programozásának ismertetése volt.

A „Megszakítás I." című programot tüzetesen áttanulmányozva arra a következtetésre jutottam, hogy az ENGED bájt programzási módszere alkalmas lehet két, váltakozva működő óra vezérlésére. Kísérletezni kezdtem, és végül sikerült a fenti követelményeknek megfelelő programot írnom.

A program a CIA#1 és a CIA#2 chip óráját működteti. Két megszakítást tartalmaz. Az egyik az órarutin aktivizálásához, a másik a KERNAL visszakapcsolásához szükséges. Betöltés után az óraprogramot SYS 49164 parancs aktivizálja. Ekkor a képernyő bal és jobb felső sarkában keretbe foglalva megjelenik a „B" és az „A" óra Ø-ra beállított kezdőértéke. Ezt követően választhatunk, melyik órát indítjuk. Az F1 gomb lenyomására az „A", az F3-ra a „B" óra indul elsőnek. A két billentyű váltakozó lenyomásával az éppen működő órát leállíthatjuk, a másikat újraindíthatjuk. A „hívott" óra a képernyőn kijelzett időértéktől számol tovább.

A SYS 49161 parancsra a működő óra leáll. A SYS 49164 ismét aktivizál, és az előbbi folyamat megismételhető.

Az óraprogram játékok készítéséhez jól használható. A SYS parancsok a játékprogramokba utasításként beépíthetők, az F1 és az F3 billentyűk lenyomása pedig szimulálható. Mint ismeretes, az éppen lenyomott billentyű kódszámát a $ C5(197) cím tartalmazza. Ezért, ha programunkban a szükséges helyekre a POKE 197,4 majd később a POKE 197,5 utasításokat beírjuk, az „A" óra fog elindulni, és a „B" órára akkor fog átváltani, amikorra terveztük.

Mindkét óra zavartalanul működik 0 és 12 óra között. Az órák időértéke programból vagy billentyűzetről állítható be. Például POKE 49152, 17:POKE 49154,89:POKE 49156,86 beírása 11 óra, 59 perc, 56 másodpercet jelez ki a képernyőn. (Az óra beállítására lásd a fent hivatkozott cikket.)

ADROVICZ ISTVÁN

```
0 REM *** KET ORA A JATEKOKHOZ ***
1 :
100 FORI=49152TO49642
110 READX:POKEI,X:S=S+X:NEXT
120 DATA 0,0,0,0,0,0,1,1,1,76,155,192
130 DATA 120,173,20,3,141,233,193,173,21,3,141,234
140 DATA 193,169,181,141,20,3,169,192,141,21,3,169
150 DATA 128,13,14,220,141,14,220,169,128,13,14,221
160 DATA 141,14,221,173,0,192,141,11,220,173,1,192
170 DATA 141,11,221,173,2,192,141,10,220,173,3,192
180 DATA 141,10,221,173,4,192,141,9,220,173,5,192
190 DATA 141,9,221,169,58,141,34,4,141,2,4,141
200 DATA 37,4,141,5,4,162,9,169,1,157,30,216
210 DATA 157,255,215,157,70,216,157,39,216,202,208,241
220 DATA 169,93,141,8,4,141,31,4,169,109,141,71
230 DATA 4,169,125,141,48,4,169,64,162,8,157,71
240 DATA 4,157,39,4,202,208,247,88,96,49,234,169
250 DATA 1,141,6,192,141,7,192,141,8,192,120,173
260 DATA 153,192,141,20,3,173,154,192,141,21,3,88
270 DATA 96,165,197,201,4,208,24,169,1,141,6,192
280 DATA 173,7,192,201,1,208,3,32,244,192,173,8
290 DATA 220,141,8,220,76,15,193,201,5,208,5,169
300 DATA 0,141,6,192,173,6,192,208,46,173,7,192
310 DATA 201,0,208,3,32,109,193,173,8,221,141,8
320 DATA 221,76,136,193,173,11,221,141,11,221,173,10
330 DATA 221,141,10,221,173,9,221,141,9,221,173,8
340 DATA 221,169,0,141,7,192,96,173,11,220,41,16
350 DATA 208,8,169,48,141,32,4,76,35,193,169,49
360 DATA 141,32,4,173,11,220,41,15,24,105,48,141
370 DATA 33,4,173,10,220,41,240,74,74,74,74,105
380 DATA 48,141,35,4,173,10,220,41,15,24,105,48
390 DATA 141,36,4,173,9,220,41,240,74,74,74,74
400 DATA 105,48,141,38,4,173,9,220,41,15,24,105
410 DATA 48,141,39,4,173,8,220,173,8,192,201,1
420 DATA 240,30,76,232,193,173,11,220,141,11,220,173
430 DATA 10,220,141,10,220,173,9,220,141,9,220,173
440 DATA 8,220,169,1,141,7,192,96,173,11,221,41
450 DATA 16,208,8,169,48,141,0,4,76,156,193,169
460 DATA 49,141,0,4,173,11,221,41,15,24,105,48
470 DATA 141,1,4,173,10,221,41,240,74,74,74,74
480 DATA 105,48,141,3,4,173,10,221,41,15,24,105
490 DATA 48,141,4,4,173,9,221,41,240,74,74,74
500 DATA 74,105,48,141,6,4,173,9,221,41,15,24
510 DATA 105,48,141,7,4,173,8,221,173,8,192,201
520 DATA 1,208,5,169,0,141,8,192,76,254,255
530 IFS<>51300THENPRINT"■HIBA A DATASOROKBAN..!":END
540 PRINT"■RENDBEN..!"
```

LASER—210

Labirintus

A négy irányjelző gomb segítségével végig kell vezetni a harcost jelképező képpontot a labirintuson, és távozni kell a kapun. Falhoz ütközés esetén a pont kimerevedik (meghal), és indul az új harcos, természetesen elölről. Négy kapun kell átjutni, amelyeket szörnyek vigyáznak. Ha a szörnyeket legyőzték, a kapuk kinyílnak. Tíz harcos élete áll egy játékos rendelkezésére.

BORZA VIKTOR

```
4 CLS:PRINT
5 PRINT "A LABIRINTUSBOL A TIZ HARCOS EGYIKENEK KI KELL JUTNI"
10 PRINT "A HALADAS IRANYA GOMBNYOMASSAL VALTOZTATHATO"
15 PRINT "A KAPUKON A SZORNYEK LEGYOZESEVEL LEHET ATJUTNI"
20 PRINT "A FALNAK UTKOZO HARCOS MEGHAL ES A KAPUK BEZARULNAK"
25 PRINT "A HALADAS SEBESSEGE 1-TOL 5-IG VALASZTHATO"
30 PRINT
35 INPUT "A KIVANT SEBESSEGFOKOZAT";A1
40 C8=125-25*A1
45 MODE(1):COLOR2
50 FOR I=1 TO 126
55 SET(1,0):SET(1,63)
60 NEXT
65 FOR I=0 TO 63
70 SET(0,I):SET(127,I)
75 NEXT :A0=-29696
77 REM VIZSZINTES FALAK
80 POKE 30862,1:POKE 30863,140
85 FOR I=1 TO 22
90 READ D
95 POKE A0+I,D
100 NEXT
105 DATA 62,3,33,00,00,17,32,0,6,00,54,85,35,27,5,32,249,25,61
110 DATA 32,240,201
115 FOR I=1 TO 21
120 READ A1,A2,A3
125 POKE A0+4,A1:POKE A0+5,A2:POKEA0+10,A3
130 A=USR(0):NEXT
135 DATA 192,112,3,197,112,4,203,112,4,209,112,6,218,112,6
140 DATA 201,113,8,215,113,3,222,113,2,192,114,2,204,114,4
145 DATA 213,114,11,194,115,5,215,115,5,222,115,2
150 DATA 196,116,3,206,116,8,222,116,2,198,117,17
155 DATA 222,117,2,136,118,7,73,119,8
157 REM FUGGOLEGES FALAK
160 FOR I=1 TO 15
165 READ D
170 POKE A0+I,D
175 NEXT
180 DATA 33,00,00,17,32,0,6,00,54,85,25,5,32,250,201
185 FOR I=1 TO 18
190 READ A1,A2,A3
195 POKE A0+2,A1:POKE A0+3,A2:POKE A0+8,A3
200 A=USR(0):NEXT
202 DATA 194,113,12,34,116,36,229,113,15,197,117,14,135,118,11
204 DATA 40,113,37,43,112,5,203,114,20,46,115,13,49,113,25
206 DATA 61,118,11,116,115,27,148,118,10,55,112,13,215,115,22
209 DATA 218,113,6,58,116,25,61,113,49
209 REM KIJARAT, ATJARUK
210 FOR K=1 TO 10
212 READ X,Y
215 FOR I=0 TO 3
220 RESET(X,Y+I)
225 NEXT :NEXT
226 DATA 24,46,32,52,33,52,67,58,120,14,120,30,120,46,126,6,126
227 DATA 22,126,38
228 RESET(127,3)
229 REM KAPUK
230 FOR I=11 TO 20:SET(I,7):NEXT
235 FOR I=107 TO 116:SET(I,56):NEXT
240 FOR I=8 TO 15:SET(22,I):NEXT
245 FOR I=41 TO 46:SET(46,I):NEXT
249 REM SZORNYEK
250 COLOR3
255 SET(4,22):SET(47,3):SET(100,32)
260 COLOR4:SET(52,24):COLOR2
264 REM INDUL A HARCOS
265 M=4:N=62
270 A=3:GOTO 420
275 REM IRANYVALTAS
280 FOR S=1 TO C8:NEXT
300 IF INKEY$="," THEN A=1
310 IF INKEY$="M" THEN A=2
320 IF INKEY$="." THEN A=3
330 IF INKEY$=" " THEN A=4
335 IF A=1 THEN 350
340 IF A=2 THEN 380
342 IF A=3 THEN 420
344 IF A=4 THEN 460
349 REM JOBBRA
350 M=M+1
357 IF POINT(M,N)>1 THEN 500
358 IF M=127 THEN 510
360 SET(M,N)
365 RESET(M-1,N)
370 GOTO 280
379 REM BALRA
380 M=M-1
385 IF POINT(M,N)<>3 THEN 390
386 GOSUB 850
387 GOSUB 400
390 IF POINT(M,N)>1 THEN 500
400 SET (M,N)
405 RESET(M+1,N)
410 GOTO 280
415 REM FELFELE
420 N=N-1
421 IF POINT(M,N)<>4 THEN 425
422 GOSUB 900
423 GOTO 440
425 IF POINT(M,N)<>3 THEN 430
426 GOSUB 750
427 GOTO 440
430 IF POINT(M,N)>1 THEN 500
440 SET (M,N)
445 RESET(M,N+1)
450 GOTO 280
455 REM LEFELE
460 N=N+1
465 IF POINT(M,N)<>3 THEN 470
466 GOSUB 790
467 GOTO 480
470 IF POINT(M,N)>1 THEN 500
480 SET(M,N)
485 RESET(M,N-1)
490 GOTO 280
495 REM HARCOS SZAMLALAS
500 G7=G7+1
502 IF G7=10 THEN 720
504 GOSUB 950
506 GOTO 230
508 REM SIKERUL A KIJUTAS
510 PRINT "  ¯¯      **** gratulalOk ****"
570 FOR I=1 TO 28
575 IF I<>15 THEN 580
576 FOR K=1 TO 150:NEXT
580 READ X,Y
590 SOUND X,Y
600 NEXT
610 DATA 23,6,24,3,26,3,26,3,24,3,23,3,21,3,19,3,19,3,21,3,23,3
620 DATA 23,5,21,1,21,4,23,6,24,3,26,3,24,3,23,3,21,3,19,3
630 DATA 19,3,21,3,21,3,21,5,19,1,19,5
640 GOTO 735
710 REM SIKERTELEN KISERLET
720 CLS:PRINT@66,"legkozelebb talan sikerul"
730 FOR I=31 TO 1 STEP-1:SOUND I,1:NEXT
735 END
740 REM KAPUNYITAS ZENEVEL
750 FOR I=107 TO 116:RESET(I,56):NEXT
760 GOSUB 900:RETURN
790 FOR I=11 TO 20:RESET(I,7):NEXT
800 GOSUB 900:RETURN
850 FOR I=8 TO 15:RESET(22,I):NEXT
860 GOSUB 900:RETURN
880 FOR I=41 TO 46:RESET(46,I):NEXT
890 GOSUB 900:RETURN
900 FOR I=22 TO 26 STEP2
910 SOUND I,1
920 NEXT
930 SOUND 31,7
940 RETURN
945 ZENE UTKOZESKOR
950 FOR I=30 TO 18 STEP-4
960 SOUND I,1
970 NEXT
980 RETURN
```

Kapcsoltam?

A televízió és a rádió vetélkedőiben gyakran előfordul olyan feladat, hogy megadnak néhány számot, és megkérdezik, hogyan folytatódik a sorozat.

Például: 1, 2, 3, ?

Az emberek többsége hajlamos ilyenkor rávágni, hogy 4 a következő szám. Ha valaki azt mondaná, hogy 10, mivel:

$1=(1-2)^3+2$, $2=(2-2)^3+2$, $3=(3-2)^3+2$, tehát: $10=(4-2)^3+2$, akkor felkészülhet egy kiadós vitára a játékvezetővel, hiszen annak más van a papírján.

A baj az, hogy az ilyen feladatoknak nemcsak egy-két, hanem végtelen sok helyes megoldásuk van! Sőt, tetszőlegesen folytatott sorozathoz találhatunk utólag képzési szabályt.

Tekintsük a megadott számokat egy függvény értékeinek az 1, 2, 3, 4, ... helyeken, amint egy sorozatot a természetes számok halmazán értelmezett függvénynek szokás definiálni (lásd az *1. ábrát*).

Nyilván a megadott pontokon keresztül végtelen sok görbét rajzolhatunk, ezek mindegyike tekinthető képzési szabálynak. Mivel mindegy, hogy hány pontot használunk fel a függvény grafikonjának megrajzolásához, ezért felvéve a nekünk rokonszenves — egyébként tetszőleges — folytatást, azon is átvezethetjük a görbét.

Egyesek talán nem elégszenek meg annak tudatával, hogy létezik az adott pontokon áthaladó görbe, szeretnék látni a függvényt képletben is. Az ő és a vetélkedők résztvevőinek kedvéért készítettem a *2. ábrán* látható programot. Az első DATA sorba kell beírni, hogy hány számunk van, a másodikba pedig a számokat. RUN-ra a program meghatározza azt a polinomot, amelynek helyettesítési értékei az adott számok az 1, 2, 3, ... argumentumokra.

Az általános Lagrange-interpoláció feladatát tehát speciális egyenlő közű esetre oldottam meg. A közölt algoritmus gyors és stabil. A megoldás a következő egyszerű meggondolásokon alapul. Ismert, hogy n ponton pontosan egy minimális fokszámú, legfeljebb n−1-ed fokú polinom halad át. Jelölje ezt $y=f(x)$ $(=a1+a2*x^1+a3*x^2+\ldots an*x^{n-1})$. A pontok az (1, y1), (2, y2), ... (n, yn) számpárok legyenek. Teljesülni kell tehát a következő egyenlőségeknek.

$f(1)=y1$ $\quad a1+a2*1^1+a3*1^2+\ldots+an*1^{n-1}=y1$

$f(2)=y2$ $\quad a1+a2*2^1+a3*2^2+\ldots+an*2^{n-1}=y2$

$f(n)=yn$ $\quad a1+a2*n^1+a3*n^2+\ldots+an*n^{n-1}=yn$

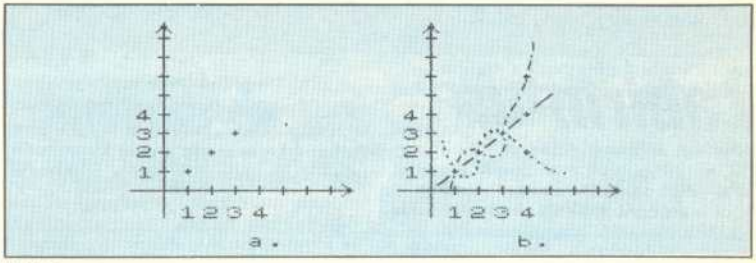
Az a1, a2, a3, ... ismeretlenekre, a keresett polinom együtthatóira tehát adott az n egyenletből álló rendszer, amely speciális szerkezeténél fogva könnyen megoldható a Gauss-féle kiküszöböléses módszerrel. A 2. ábrán közölt program még ezt is egyszerűsíti, rögtön a kiküszöböléses eljárás eredményeként létrejövő háromszög alakú egyenletrendszert írja fel és oldja meg.

A *3/a, b, c ábrán* látható futási eredmények az 1, 2, 3, 4, az 1, 2, 3, 10 és az 1, 2, 3, 22 sorozatokhoz készült interpolációs polinomokat mutatják. Bármely számmal is folytatunk egy sorozatot, a program megmagyarázza választásunkat!

A számtalan függvénytípus közül egyszerűségük miatt a polinomokat választottuk. Ebben az értelemben teljesítettük azt a meg nem fogalmazott kikötést is, hogy a képzési szabály ne legyen bonyolult. Ha valakinek az egyszerűség itteni kritériuma nem felel meg, gondolkodjon kissé, s rájön, hogy az egyszerűség meghatározása milyen bonyolult.

Mindebből az következik, hogy a „Hogyan folytatódik ..." kérdésre a legprecízebb válasz: „Bárhogy!".

LOVRICS LÁSZLÓ



1. ábra

```
10>PRINT "Values for interpolation :"
   PRINT
20 READ n
30 DIM g(n)
   DIM a(n,n)
40 FOR j=1 TO n
50   READ g(j)
60   PRINT j,g(j)
70   LET a(1,j)=1
80   FOR i=2 TO j
90     LET a(i,j)=i*a(i,j-1)+a(i-1,j-1)
100    LET g(j)=(g(j)-g(i-1))/(j-i+1)
110  NEXT i
120 NEXT j
130 PRINT
   PRINT "The interpolated polinom :"
   PRINT
140 FOR i=n TO 1 STEP -1
150   FOR j=i+1 TO n
160     LET g(i)=g(i)-a(i,j)*g(j)
170   NEXT j
180   PRINT g(i),"*x^";i-1
190 NEXT i
200 DATA 4
210 DATA 1,2,3,22
```

2. ábra

3/a ábra

```
Values for interpolation :

1   1
2   2
3   3
4   22

The interpolated polinom :

3      *x^3
-18     *x^2
34     *x^1
-18     *x^0
```

3/b ábra

```
Values for interpolation :

1   1
2   2
3   3
4   10

The interpolated polinom :

1      *x^3
-6      *x^2
12     *x^1
-6     *x^0
```

3/c ábra

```
Values for interpolation :

1   1
2   2
3   3
4   4

The interpolated polinom :

0      *x^3
0      *x^2
1      *x^1
0      *x^0
```

Nagyon sok olvasónk küld géppel vagy kézzel írott programokat. Ezeket sajnos nem mindig tudjuk letesztelni, mert arra nincs lehetőségünk, hogy három-négy oldalas programokat gépbe billentyűzzünk. Akik azt szeretnék, hogy programjaikat közöljük, azok ne adathordozón küldjék el munkájukat, hanem nyomtatott és *nem* géppel vagy kézzel írott programlistán (egy sorban max. 40 karakter), mert csak így tudunk elfogadhatóan tördelt oldalakat szerkeszteni. Előre is köszönöm az olvasók szíves megértését és aktivitását.

Ifj. Szadai József, Nyékládháza,

Kandó K. út 14. 3433

Ne haragudjon, hogy levelemmel zavarom, de van néhány olyan problémám, amelyre nem tudok választ kapni, de remélem, hogy Maga, vagy ismerősei meg tudják magyarázni azokat a dolgokat, amelyet tapasztaltam.

Negyedik osztályos tanuló vagyok, a leninvárosi Kun Béla középiskola vegyész szakán tanulok.

Júliusban kaptam egy COMMODORE 64-es számítógépet, hozzá tartozó magnóval. Előtte már jól megtanultam a BASIC programnyelvet és a 6510-es mikroprocesszor assembly nyelvét.

Édesapám munkatársától kaptam három jó programot: a PROFI—ASS 64-et, a SUPERGRAPHIK 64-et és a SIMON'S BASIC-et. Sajnos az örömöm nem sokáig tartott: a SIMON'S BASIC program jól működött, de a másik kettő csődöt mondott.

A PROFI—ASS 64 assemblerről azt olvastam, hogy magnóval is működik, nem kell hozzá lemezegység. A betöltés után azonban, amint egy sort beütöttem, az „OUT OF MEMORY ERROR" hibaüzenetet adta. FRE(0)-val lekérdeztem a szabad byte-ok számát, amelynek értéke —4 lett. Sokat próbálkoztam, de eredménytelenül.

A másik programnál más volt a baj.

A betöltő programot betöltöttem és RUN-nal elindítottam. Törölte a képernyőt és kiírta a READY-t. A betöltő programnak kellett volna betölteni a másik két főprogramot, a GRAPHIK IEC-t és a GRAPHIC NOR-t. De sajnos nem történt semmi. Akitől a programot kaptam, megkérdeztem, hogy mi lehet a baj. Véleménye szerint az a baj, hogy mivel neki lemezen van ez a két program (a PROFI—ASS 64 és a SUPERGRAPHIK 64), lehet hogy az a baj, hogy a betöltő programban a 8-as lemezegységszám van, és erre, pontosabban a lemezegységre vonatkozik, szemben a magnó 1-es azonosító számával. Szerinte tehát azokat a tárcímeket, amelyen a lemezegységre vonatkozó 8-as szerepel, 1-esre kell átírni, és akkor működni fog. De szerintem, ha így lenne, akkor a „DEVICE NOT PRESENT" hibajelzést kéne hogy adja, mivel a lemezegység nincs a géphez kapcsolva.

Mivel tudomásom szerint többféle SUPERGRAPHIC 64 program van, leírom, hogy az enyém, amennyiben működne, hogy jelentkezne be:

SUPERGRAPHIK CBM 64
(C) 1983 BY DATA BECKER
ENTWICKELT ZUM 03. 10. 1983
VON AXEL PLENGE — MEERBUSCH

Várom azoknak a Commodore tudós olvasóinknak a jelentkezését, akik vállalkoznak a távdiagnosztikára.

Gutai Zsolt, Jászberény,

Hunyadi J. u. 30. 5100

A nyár végén hozzájutottam egy Commodore 128-as típusú számítógéphez, azonban az a hiba, hogy a gépkönyv német nyelvű. Mivel sem én, sem barátaim nem tudnak ilyen szakszöveget leforditani, ezért olvastam egy régebbi (1985/5.) µ M számában örömmel azt a levelet, amelyben a Babilon Nyelvstúdió írja az olvasóknak, hogy vállalkoznak kezelési utasítások magyar nyelvű leforditására is. Ez így rendben is volna, csakhogy miután feladtam a levelet a Babilon Nyelvstúdió részére, kb. 1 hét múlva visszajött a levél, mert a címzett a Posta által ismeretlen helyre költözött címszóval a posta visszahozta a levelet.

Ezért kérném az Ön segitségét, hogy ha lehetősége van rá, az alábbi boritékra ráirva a nyelvstúdió címét, a részükre elküldeni szíveskedjék.

Elküldtük.

Halmai Zsolt, Budaörs

Kassai u. 3. 2040

Minden hónapban a megjelenés időpontjában megveszem a Mikroszámítógép Magazint. Az árával semmi problémám. Érdekes, szerteágazó, széles körű, jól megszerkesztett lap. Én magam túl vagyok a tízentúlon. Alig két éve foglalkozom számítástechnikával, de eléggé kezdetleges módon. A számítógépet egy BASIC-tanfolyamon ismertem meg közelebbről. Primón tanultam, de már akkor beleszerettem a Spectrumba. Barátom unszolgatására és jómagam ösztönzésére ez év májusában vásároltam is egyet. Azóta írtam két kalandos témájú programot, és kezdettől a programozást kedvelem. Játszani csak pihentetőül szoktam.

A Magazint olvasva sok érdekes dolgot olvashatok, találok. Több olvasóval egyetértésben én is keveslem a Spectrummal kapcsolatos programokat, szoftver ismertetőket, hardver ötleteket. „Commodore centrikusság" jellemző a lapra. A gépi kódú ismertető cikk is Commodore gépre íródott. Sajnos!

Szeretnék a lapban több Spectrum-cikket olvasni.

Folyamatosan kapjuk a Spectrum programokat kevesló leveleket. Lassan azt hiszem, hogy a Spectrum-tulajdonosok csak olvasni szeretnek, de programot írni nem, míg a Commodore-tulajdonosoknál a helyzet egészen forditott. Küldjenek jó Spectrum programokat, közölni fogjuk.

Dr. Tikász Attila, Berettyóújfalu,

József A. u. 5. 4100

A 10. számban olvastam Képiró Róbert levelét, illetve az erre adott választ a Spectrum-tulajdonosokkal kapcsolatban. Hasonló tartalmú levél az Ötlet szeptemberi számában is megjelent. Ott is hasonló volt a szerkesztőség reagálása.

Mint „Spectrumos" szintén csatlakozni tudok a levélírókhoz, és nem tudom maradéktalanul elfogadni a szerkesztőség válaszát.

Akárhogyan nézzük, a Spectrum a másik 3 gép (Commodore, HT és Primo) mellett a legelterjedtebb; ugyanakkor nem dúskálunk nagyon az irodalomban. Bárhogyan is tagadnánk, a Commodore favorizált helyzetben van a többi géphez viszonyítva. Elég bemenni egy nagyobb könyvesboltba és szemrevételezni a számítógépes irodalmat. Egyik kezünkön meg lehet számolni, menynyi könyv foglalkozik a Spectrummal (a HT és Primo még rosszabb helyzetben van). E könyvek színvonala sem olyan, mint a Commodore-é. Nem sokkal mondanak többet, mint az eredeti gépkönyv.

Én úgy érzem, hogy a spectrumosoknak nem az fáj, hogy kevés a programlista akár az Önök újságjában, akár az Ötletben vagy máshol, hanem kevés a szakirodalom, egyét részletkérdéssel foglalkozó komolyabb szakcikk. Akár a 10. számot, akár a korábbi számokat nézzük, erről azonnal meg lehet győződni.

Azt hiszem, valamennyi spectrumos örülne, ha egy olyan könyvsorozatot vehetne kézhez, mint például a Data-Becker sorozat a Commodore-hoz. Külföldön van ilyen színvonalú irdalom akár német, akár angol nyelven. Egyszer jó lenne gondolni a Spectrum-tulajdonosokra is egy ilyen sorozat kiadásával.

De Önök is segithetnének. Például nagyon elképzelhetőnek tartanám, ha megjelenik egy komolyabb cikk a HT alkalmazásával kapcsolatban (éppen a Z80-as processzorra tekintettel), közölnék azokat a változtatásokat, amelyek elvégzésével a Spectrumon is használható lenne a cikkben említett megoldás. Ezt az olvasók többsége nem tudja otthon megcsinálni — éppen a szakirodalom (HT) hiánya miatt.

Meggyőződésem, hogy ilyen kis segítségnek is örülnének a spectrumosok, éreznék, hogy nem annyira mellőzöttek. (Én sem vagyok előfizetője lapjuknak, éppen Képiró Róbert levelében írtak miatt. Esetenként megveszem a lapot, amikor Spectrumot vagy esetleg HT-t érintő cikket találok benne.)

Lenne ellenben két javaslatom, biztosan örülnének neki a spectrumosok. Egy cikksorozatban jó lenne ismertetni, részleteseben írni a Spectrum rendszerváltozóiról, alkalmazásának lehetőségéről. Az Ipari In-

formatikai Központ ROM-könyve is csak a 16 384. címig foglalkozik a géppel. Nagyon kevés ezekről a részletes információ. Ehhez kapcsolódna a másik javaslat: egy összehasonlító cikk vagy táblázat a HT és a Spectrum rendszerváltozóiról. A fentebb említett „átírásokhoz" segítséget nyújtana.

Ismét a Spectrumról. Elgondoltam a levelén, hogy miért is nincs elég szakirodalom ehhez a géphez, azt hiszem rájöttem a megoldásra. A Sinclair gépek minden darabja magánimport útján került az országba. A kiutazó turisták azért vették meg ezt a gépet, mert kb. feleannyiba került, mint a hasonló tudású Commodore. Sinclair gépet itthon legfeljebb bizományiban árultak, ennek a gépnek sohasem volt hivatalos importőre.

Nem úgy a Commodore-nak. Elsősorban a Novotrade, de az Ápisz is és mások is importálták. Professzionális kereskedőkről lévén szó, jól tudták, hogy kiváló gépkönyv, alkalmazói kézikönyv és szoftver leírás nélkül a hardvert eladni nem lehet. Érdeküket volt és ma is az a könyvkiadás. Talán szükség lenne hasonló és pénzes vállalkozóra, aki megkockáztatná néhány könyv kiadását. A Műszaki Könyvkiadó és a μ M ezt egyszer megtette, 1985-ben kiadtuk a Votisky Zsuzsa szerkesztette „Csupa játék ZX Spectrumra" c. könyvet. Egyelőre újabb Spectrum könyvre nem vállalkozunk.

Honti József, Csákvár,

Május 1. út 11. 8083

Hosszú hallgatás után jelentkezem ismét. E hosszú szünetnek részemről több, igen alapos oka is volt. Hadd kezdjem az elején!

Augusztus végén immár másodszor rendeztünk számítógépes bemutatót községünkben. A bemutatónak egyértelműen nagy sikere volt. 30 db számítógépet sikerült összeszedni, ami már magában sem kis teljesítmény. A gépek döntő része Commodore típusú volt, elsősorban 64-esek. Szerepelt a bemutatón PRIMO és természetesen itt volt a VIDEOTON TVC gépeivel. Az idén nemcsak a laikus érdeklődőkre számítottunk, hanem azokra is, akik már rendelkeznek géppel, vagy legalábbis hozzájutnak a számítógéphez. Ezért a bemutatónkat a mikroszámítógép barátai első Fejér megyei találkozójának is szántuk. Sok meghívót küldtünk szét a kluboknak és sokan elfogadták meghívásunkat. Az érdeklődők között volt budapesti, hatvani, salgótarjáni és berettyóújfalui illetőségű is.

Kérésének — hogy a találkozóról írjak beszámolót az újság részére — azért nem láttam értelmét eleget tenni, mert rendkívül hosszúnak találom a nyomdai átfutási időt, és mire megjelenne a cikk, már régen nem aktuális a téma. Egyébként úgy vélem, hogy sokkal jobban szolgálom ezt az általam önként felvállalt ügyet, ha itt, Csákváron teszek meg minden tőlem telhetőt a számítástechnikai ismeretek terjesztéséért.

Hosszú hallgatásom másik oka az volt, hogy egyik barátommal egy ún. Klublemez összeállításon dolgoztunk. Ez a lemez a C16-os gép lehetőségeinek jobb megismerését és hatékonyabb kihasználását kívánja elősegíteni. Tulajdonképpen igyekeztünk minden olyan alapismeretet összegyűjteni a hazai és külföldi folyóiratokból, ami segíti az evvel a géptípussal folyó munkát. Azt, hogy sikerült-e ezt a célt elérni, vagy legalább megközelíteni — döntse el más, aki ebben nálunk avatottabb.

A lemez dokumentációját mellékelten megküldöm. Érdeklődés esetén — eszmei tiszteletdíj ellenében — szívesen rendelkezésükre bocsátjuk és átengedjük a terjesztés jogát. Mindenképpen szeretnénk véleményt kérni a munkánkról, hiszen az segítséget és iránymutatást adna a továbbiakhoz is.

Örömmel tájékoztatom arról, hogy községünk iskolája 12 db C—116-os gépet kapott. Így a gyerekek a jövőben könnyebben hozzájutnak a számítógépekhez úgy, hogy a használatáért semmilyen címen nem kell fizetniük. Az iskola vezetésének előrelátását dicséri, hogy már az elmúlt tanév folyamán a nevelők közül a megyeszékhelyeken szervezett tanfolyam keretében felkészítették a gyerekek oktatására. (Kár, hogy csak egy főt!)

A művelődési ház kezelésében levő 6 db számítógép és a most kapott 12 új gép már olyan komoly gépparknak számít, amire két évvel ezelőtt még álmomban sem mertem gondolni.

Fentiek ismeretében a logikus lépés az lenne, ha a két gépparkot összevonnák. Sajnos erre különböző okokból nincs lehetőség. A Művelődési Ház továbbra is ragaszkodik a saját gépeihez, az iskola vezetése meg úgy látja, hogy a rendelkezésre álló gépeket sem fogja tudni megfelelően üzemeltetni. Én azt hiszem, hogy mégis csak az összevonás lenne helyes lépés úgy, hogy az iskola adna otthont a felnőtt oktatásnak is. (Persze ezekhez a dolgokhoz nem sok közöm van, és beleszólási jogom sincs.)

Köszönöm a beszámolót; ha valahol a település összefog az informatika társadalmasítása érdekében, akkor ott csodákat lehet tenni. Csákvár példája ezt mutatja, de azt is, hogy az összefogásnak is vannak fokozatai és azon is lehet még javítani.

Járóka László, Budapest,

Szív u. 3—5. fszt. 5. 1063

Egyik lapjukban olvastam az Internacia Komputado lap hirdetését, de a hirdetésből az derült ki, hogy 92 országban jelenik meg. Szeretném felhívni a figyelmüket, aki nem ismeri a lapot, az nem tud róla semmit, hogy hol és milyen nyelven. Én tudom, és minden száma megvan. Itt szeretném felhívni a figyelmüket egy Hollandiában működő DLT-rendszerre, aminek az eszperantó nemzetközi nyelv az alapja. A DLT-rendszerről kellene egy kis cikk, amivel ismertetni kellene, mit is jelent ez. És ha sikerült, akkor egy számítástechnikai nyelvet, mivel a — DLT — egy fordító rendszer és az alapja az „Eszperantó nyelv". Mivel a szerkesztőségben van egy pár eszperantista, és a nyelvi problémák hamarabb megszűnnének.

Remélem, hogy a levelem megfontolásra kerül és nem szemétkosárba.

Nem szoktuk a leveleket a szemétkosárba dobni. Örülnénk viszont, ha tájékoztatna bennünket, esetleg cikket is írna a DLT-rendszerről.

Sebestyén László, Budapest,

Lévai Oszkár u. 12. 1084

Jelenleg egyetemi hallgató vagyok és sokat foglalkozom számítástechnikával. Lapjuknak rendszeres olvasója vagyok, az első évfolyamtól az összes szám megvan. Nekem egy C64-es számítógépem van, ezért főleg a C64-es programok érdekelnek. Nagyon tetszett a KERNAL rutinok ismertetése, ámbár tapasztaltam egy-két hiányosságot. Nagyon hasznos lett volna megemlíteni, hogy C64-nél a LOAD rutin nem működik, ha használatával egyazon időben sprite-okat is használunk. A sprite megjelenítése a processzor idejéből sok óraciklust elvesz, így a soros buszt nem tudja szinkronizálni. Ezért programbetöltésnél a sprite-okat ki kell kapcsolni. Nagyon sok fáradságomba került, mire rájöttem, ezért kérem, hogy mielőbb publikálják ezt a kiegészítést, más ne fáradjon hiába.

1986. februártól nagyon tetszik a sakkprogramozás rovatuk is. Kovács P. Attila nagyon jól bemutatja a sakkprogramok felépítését. Ámbár úgy gondolom, hogy a Bitek és figurák sorozat összegyűjtve brosúraként, vagy könyv formátumban is megjelenhetne, mert a havonta megjelenő számokból nagyon keveset tud meg egyszerre az olvasó, pedig aki ilyen programot szeretne írni, annak az algoritmus globális áttekintésére is szüksége van. Én szívesen írnék sakkprogramot, ezért egy-két problémámmal meg szeretném keresni a cikksorozat szerzőjét. Kérem segítsenek! Kérem, írják meg azt is, hogy az NJSzT által kiírt sakkprogramozási versenynek mi lett a végeredménye!

Köszönöm a tanácsot. Sakkprogramozás ügyben a téma magyar „pápáját", dr. Lindner Lászlót, sakkrovatunk vezetőjét tudja felhívni (tel.: 318-096).

Gruber László, Budapest,

Buday László u. 7. III. 1. 1024

. . . A levélért különben is csak saját magukat „okolhatják", mivel levelezési rovatuk elején erre biztatják a „kedves olvasó"-t (Hát érdemes volt?)

1. A lapot egészében nézve igen jónak és megfelelő színvonalúnak tartom, az első számtól kezdve gondosan gyűjtöm. Néhány észrevételem azonban lenne.

1.2. Némi törtelési megjegyzés. Feltételezve néhány olvasónál azt a „szentségtörést", hogy házi programkönyvtára érdekében — ne adj isten — ollóval merészel a laphoz nyúlni, néha igencsak bajba kerül. Ha pl. teszem azt a 86/10. számnál külön akarná választani a 25. oldalon kezdődő C64 programtitkosítást és a 27. oldali ZX sprite rajzolást, akkor bajban van, mert vagy az egyiket, vagy a másikat teszi tönkre.

Ráadásul a 28. oldal alján levő „programlista kódok" olyan táblázat, melyet ez-

után mindig használni kell, tehát központi hely „illeti meg". De nem lehet kivágni, mert akkor mindkét említett cikk károsodik.

Nos a javaslat: Úgy kellene tördelni a lapot, hogy a programok, vagy táblázatok „hátuljára" olyasmi kerüljön, amiért nem kár. Pl. hirdetés, vagy a 2., illetve a 24. oldali cikk, de lehetne a 22. oldali is, vagy — horribile dictu — a 43., 44. oldalak is.

Lehet, hogy így „széttördelődnének" a rovatok, de ez az olvasót vajmi kevéssé érdekli, az előző szempont — a „szétválaszthatóság" — szerintem sokkal fontosabb lenne.

2. „Örökzöld" téma az egyes számítógépek inkompatibilitása, az egyikre írt programok nem futtathatók a másikon. Nagy gyakorlattal és hozzáértéssel persze „átírhatják", de ez nem túl egyszerű feladat.

Úgy tűnik, a probléma nem megoldhatatlan. Nekem pl. van egy programom (floppy lemezen 104 block), amelyet C64-be betöltve és lefuttatva, a C64-et „átalakítja" ZX Spectrummá (a „bejelentkezés" is ennek megfelelő lesz) és a DATASETT-be (1531) minden további nélkül betehető egy ZX programkazetta, az behívható és futtatható.

Úgy gondolom, ha ez két ilyen különböző gép esetén megoldható, akkor 2 Commodore között is.

Óriási dolog lenne egy olyan program közlése, mely a +4-be betöltve az fogadni tudná a C64 programokat!!!

Fordítva már nem tudom elképzelni a dolgot, mivel a +4 többet tud a C64-nél, így a „tanítás" nehéz lenne.

Bizonyára nem egyszerű dolog ilyen program írása, s nem tudom, hogy a szerző kellően lenne-e dotálva a közlés díjával, de én az hiszem, ha csak pénzért lehetne vásárolni, akkor is minden +4 tulajdonos boldogan megvenné, feltéve persze, hogy észszerű áron (max. 1-2 E Ft) adnák.

3. S végül egy apróság. A C64 perifériái jók a +4-hez is. Olyannyira, hogy pl. a +4-hez árult botkormánynak C64 csatlakozója van, s külön átalakító kábelt adnak hozzá, hogy a +4 DIN csatlakozója „fogadni" tudja. Csak a DATASETT-re nem érvényesek fentiek. Ráadásul, míg a C64 gépkönyve részletesen ismerteti az 1530-as bekötését, addig a +4 gépkönyve erről mélyen hallgat.

A kérésem tehát: Ha lehetséges, közöljenek egy kis bekötési rajzot, mely egy kábel egyik végén egy DIN aljzatot (az 1531-eshez illeszkedőt) míg a másik végén egy C64 DATASETT csatlakozót tartalmaz, s melynek használatával az 1531-es magnó használható lenne C64-hez is.

Olvasónk leveléből részleteket közöltem, ezenkívül javaslatot tett, hogy hirdetések helyett inkább árat emeljünk, elmondta, hogy miért nem való a lapba a Teleteaching '86-ról szóló írásom (sajnálom, hogy a cikkből nem derült ki, mi is a kapcsolat a távoktatás és a mikroszámítógépek között) és megjegyzéseket tett a Novotrade árairól. A fentieket éreztem a levélből közérdekűnek, el is gondolkoztunk a javaslatokon; a bekötési rajzzal kapcsolatban pedig a Commodore Klubhoz fordulunk.

Leveleiket továbbra is várja:

KOVÁCS GYŐZŐ

Jánoki Lajos—Kocsis János: Számítógépes termelésirányítás (Budapest, 1986. Műszaki Könyvkiadó, 329 oldal. Ára: 90,— Ft.)

A termelés- és gyártásirányítás — különösen az utóbbi évek egyre bonyolultabbá váló gazdasági helyzetében — az ipar és a mezőgazdaság különböző ágazataiban mind több szakembert érint. Az iparvállalat mint irányított rendszer, hierarchikus rendszer, amelynek a termelésirányítás az alsó szintjét, azaz az ún. alapszintjét képezi, ellátva a legfontosabb műszaki-gazdasági információs feladatokat, és ezzel megteremtve a felsőbb szintű döntések objektív alapját.

A könyv elsődleges célja, hogy megismertesse az olvasóval a termelés- és gyártásirányítás bonyolult részfeladatait (gazdaságos gyártás-előkészítés, anyagellátás, operációkutatási feladatok, üzemirányítás, termékértékesítés, ezek információs rendszerei stb.), valamint a részfeladatok egymáshoz való kapcsolódását a hierarchikus irányítási rendszeren belül, a legkisebb területekre is kiterjedően. Így hatékony segítséget nyújt a vállalatok, termelő üzemek minden irányítási szintjén érintett vezetőknek, mérnököknek, közgazdászoknak, matematikusoknak.

Külön fejezetekben foglalkozik a vállalatvezetési információs rendszerrel, a számítógépes termelésirányító rendszer felépítésével, a termelésirányítás hardvereszközeivel, a termeléstervezéssel és készletgazdálkodással és végül bemutat egy megvalósított, hazai termelésirányító és anyaggazdálkodási rendszert.

Prague, Cary N.— Hammitt, James E.: Adatbázis-kezelés. Programozás. dBASE III. (Budapest, 1986. Műszaki Könyvkiadó, 273 oldal. Ára: 98,— Ft.)

A dBASE III hatékonyan alkalmazható adatbázisnyelv. Felhasználóközeli utasításokat tartalmaz, amelyek adatbázis-szerkezetek létrehozására, valamint adattételek hozzáadására, módosítására, kiválasztására és törlésére alkalmasak. A dBASE III az adatbázisban lévő rekordok keresését, kiválasztását és megjelenítését is lehetővé teszi. Segítségével könnyen lehet jelentéseket tervezni és kinyomtatni. A mikroszámítógépek egyre táguló világában gyorsan nő e programnyelv népszerűsége.

A dBASE III „teljes" programnyelv, amely alkalmas csaknem valamennyi adatfeldolgozási feladat megoldására. Az adatbázis-technika használata miatt a dBASE III egyszerűbbé teszi a programozást, mint amilyen az a hagyományos nyelvek esetén.

A könyv nemcsak az újdonsült programozókhoz szól, hanem a tapasztaltakhoz is, akik új nyelvvel akarják bővíteni eddigi ismereteiket. Feltételezi az olvasóról, hogy létrehozott legalább egy adatbázist, és előállított már néhány egyszerű jelentést. A kezdő programozót a programozási alapelvekbe, az adatbázis-technikába is bevezeti. A gyakorlott programozó megfelelő útmutatót talál arra az esetre, ha az általa már ismert bármely nyelvről a dBASE III nyelvre akar áttérni.

Dahmke, M.: Mikroszámítógépek operációs rendszerei (Budapest, 1986. Műszaki Könyvkiadó, 199 oldal. Ára: 79,— Ft.)

A kötet szerzője az alapoktól indulva, a szoftver oldaláról közelítve mondja el mindazt, amit a mikroszámítógépek operá-

ciós rendszereiről a felhasználónak tudnia érdemes és kell. A legfontosabb hardver-, technológiai és szoftverfogalmak tisztázása után részletesen szól a kis, közepes és nagy rendszerek jellemzőiről. Kitér a többfelhasználós rendszerekre, a hálózatokkal, a memóriagazdálkodással, a gépfüggetlenséggel kapcsolatos ismeretekre. Az olvasó megismerkedhet több konkrét operációs rendszerrel is.

A függelékben rövid leírás található a CP/M, az MP/M és a UNIX operációs rendszerről.

Bakos Tamás:
Pascal PC-seknek
(Budapest, 1986.
Műszaki Könyvkiadó,
161 oldal. Ára: 53,— Ft)

A könyv ismerteti a személyi számítógépeken is mindjobban terjedő, a BASIC helyébe lépő programozási nyelv, a Pascal elemeit, jellemzőit és a nyelv alkalmazásától elválaszthatatlan programozási módszereket. Minthogy a BASIC nyelv ismeretére alapozva, azzal való folyamatos összehasonlítással vezeti be a Pascal-beli új fogalmakat, a személyi számítógépek felhasználói BASIC-tudásuk alapján gyorsan elsajátíthatják az új nyelvet.

Útmutató
a számítástechnika
alkalmazására
vonatkozó érvényes
jogszabályokról
és irányelvekről
(Készítette: Szabó József
és Borda József)
(Budapest,
SALDO Pénzügyi Szervező
és Tanácsadó Vállalat,
1985. (1986.) 203 oldal.
Ára: 362,— Ft.)
A Pénzügyminisztérium
hivatalos kiadványa

A kötet tartalmazza a számítástechnika területére vonatkozó pénzügyi, számviteli, ellenőrzési, árképzési, installálási, aktiválási, elszámolási, archiválási, adatvédelmi, adatbiztonsági szabályozásokat és magyarázatukat, mind az ágazatért felelős KSH, mind pedig a bizonylati rendet szabályozó és a működtetést, alkalmazást ellenőrző PM előírásainak megfelelően. A kötet összeállítóit az a cél vezette, hogy a számítástechnikai ágazatokban alkalmazandó joganyagot fogalommagyarázatokkal ellátva, külön kiadványként adják közre, megkönnyítve ezzel a gazdálkodó és az ellenőrző szervek egységes szemléletű jogalkalmazását.

Az ismeretlen C16

Erdős Iván: COMMODORE
ROM lista
PLUS/4, C—16, C—116
(Budapest, 1986.
LSI ATSZ,
281 oldal. Ára 248,— Ft.)

Kérdés, hogy helyesen választottam-e ki a könyv címét, ugyanis a borítón és a belső címoldalon a cím négy különböző változata szerepel. Nekem ez volt a legrokonszenvesebb. A könyv az LSI „csíkos" sorozathoz tartozik, és magán viseli e sorozat korábbi Commodore tárgyú köteteinek ismertető jegyeit: a sok értékes új információt a hibák olyan tömegével, amely az információk használhatóságát kétségessé teszi. Nem tudom, hogy mit higgyek el, mit ne.

A könyv legnagyobb részét a közel 32 kilobájtnyi disassemblált ROM lista foglalja el. Bár a szerző több alkalommal leírta, hogy ez a lista a PLUS/4 ROM-jából készült, én valószínűbbnek tartom, hogy a másik két géptípus valamelyikéből. Egészen pontosan: jobban hasonlít a lista a C16 belső felépítésére, mint a PLUS/4-ére, de nemcsak az alkalmazói programok — a bevezetőben megindokolt — hiánya miatt.

A Műhelytitkok dióhéjban című részben megismerkedhet az olvasó a könyv elkészítéséhez kijelölt alapelvekkel és megvalósításuk módjával. Megtaláljuk ebben a részben a könyv készítéséhez használt segédprogramok forráslistáját is, a PLUS/4-re (?) írt disassembler program bőségesen el van látva közbeiktatott magyarázatokkal. A programok szövege egy kicsit nehezen olvasható, látszik, hogy nem közlés céljára íródott.

A könyv függelékében keresztreferencia táblázatot találhatunk. Zavaró, hogy a hivatkozott címek rendezettek ugyan, de a hivatkozási helyek címei teljesen össze vannak keverve egy-egy hivatkozáson belül. A táblázatban sok a hamis hivatkozás; ezek többsége egy kicsit gondosabb munkával kiszűrhető lett volna. Olyan hibákra gondolok, mint kétszeri hivatkozás a $FFA9 címre. Viszont nehéz elkerülni az olyan problémákat, mint például a sztring veremre történő hivatkozások egy része, mely látszólag — és a keresztreferencia lista szerint — a $00 címre szól, a valódi címet az X regiszter tartalmával való indexelés adja.

Másik hibája a keresztreferencia listának, hogy sok olyan cím nem szerepel benne, melyre a ROM lista szerint van hivatkozás. Ilyenek többek között a táblázatokban elhelyezett címek.

A ROM lista készítésének egyik alapelve volt, hogy kizárólag hexadecimális számokat használt a szerző, a $ jeleket elhagyva. Jó lett volna ehhez az elvhez ragaszkodni a magyarázó szövegek írásánál is, ahol decimális számok ugyanúgy találhatók, mint hexadecimálisak, ez utóbbiak néha $ jellel, néha anélkül. A magyarázó szövegek másik hibája a tömörségre való törekvésből ered. Néha érthetetlenek, az is előfordul, hogy félrevezetők. Példa erre a parancsmód ellenőrzését végző szubrutin hívása, ahol a „'direct mode only'", illetve a „'direct mode only', ha" magyarázó szöveget olvashatjuk. Másik példa a 206. oldalról: a „Monitor hívása" és a „Monitor break" címekből nehezen lehet kitalálni, hogy tulajdonképpen miről is van szó. Az ilyen apróságoktól eltekintve a magyarázatok jók, és kitűnően eligazítják azokat, akik félszavakból is értenek.

Nagyon jó — bár nem teljesen eredeti — ötlet, hogy Lxxxx alakúak azok a címkék, amelyekkel a szerző megjelöli a programnak azon pontjait, melyekre valahonnan hivatkozás jöhet. Elvétve így is előfordulnak hamis, illetve hiányzó címkék, de ez a feldolgozás módjából eredően elkerülhetetlen.

A feldolgozás egyik legnehezebb része az utasítások és az egyéb információk (táblázatok, ASCII szövegek, kódok) szétválasztása. Ez az esetek túlnyomó részében kitűnően sikerült, de itt is előfordult melléfogás. Jó példa erre a Lab88 címkével kezdődő három sor, melyek az előzőleg felsorolt BASIC tokenek disassemblálásából keletkeztek, a tokeneket gépi kódnak vélve.

A 168. oldalon furcsa hibára akadtam. Az indirekt indexelt címzési mód y-ja a hozzá tartozó vesszővel együtt a zárójelek közé került. Nem értem, hogyan történhetett ez meg.

Összegezve: akit érdekel a C16-nak és családjának lelkivilága, és nem új neki ez a téma, sok hasznos információt megtalálhat ebben a könyvben. Én is sok munkától megszabadulam volna, ha a könyv kb. 10 hónappal korábban megjelenik. Mégis azt ajánlom, hogy aki teheti, válassza a Novotrade Rt kiadásában megjelenő, hasonló tárgyú Data Becker könyvet, amelyről e sorok írásakor csak annyit tudok, hogy várható a megjelenése, de remélem, mire ez a cikk az olvasó elé kerül, már kapható lesz. Ha mégsem, akkor is érdemes megvárni.

BARNA LÁSZLÓ

Sakkozó Elefánt

Aki híve saját sakktudása továbbfejlesztésének, vagy szeretné gyermekének lehetővé tenni, hogy önállóan megtanuljon sakkozni, az remek tanítómesterhez jut a Pintér József nemzetközi nagymester közreműködésével készült Sakkozó Elefánt nevű program alkalmazásával. Ez a sakkpedagógia legújabb eredményeit felhasználva, módszeresen összeállított feladványokat ad, segítséget nyújt a megoldáshoz, és rendre értékeli a tanuló tudását. Biztosítja az alapvető sakkszabályok tökéletes elsajátítását, a taktika és stratégia megismerését s a tanultak gyakorlását a beépített játékprogram segítségével.

A programcsomagot az Economix Közgazdász Egyetemi Kisszövetkezet forgalmazza a lemezmeghajtóval ellátott Commodore 64 és 128-as gépekre.

Olcsó PC

Csehszlovákiában a zsolnai VÚVT 1984-ben mutatta be négy gépből álló mikroszámítógép-családját, köztük az SP—01 jelű háziszámítógépet. Az utóbbinak a billentyűházába IBM PC-vel kompatibilis gépet építettek, melyet PP—01.16 néven mutattak be. E gép külön érdekessége, hogy ez a Szovjetunión kívül az első alkalmazása az új szovjet mikroprocesszornak, az Intel 8088-nak megfelelő KM 1810 VM88 típusnak. A gép ára meghökkentően alacsony: a billentyűzettel együtt csupán 25 ezer korona (55 ezer forint), bár a tervezett gyártási sorozat túlságosan kicsi; az idén kétszáz darabot, jövőre pedig hétezret kívánnak előállítani.

Szovjet szakértő vizsgálja a PP—01.16-ot

Radiosoft

Jugoszláviában már 1984 óta sugároznak számítógépes programokat rádióadásokba építve. Ennek eredményeképpen a hazánk déli körzeteiben élő Commodore 64- és Spectrum-tulajdonosok jó néhány programmal gazdagodhattak. Ehhez csupán jó minőségben kazettára kellett venniük az adást, amihez egy közönséges rádiós magnetofon is elegendő felszerelés volt, s máris tölthették a friss programot a gépbe. Az „elhangzott" programokat Budapesten már csak komoly antennával lehet venni, hiszen a minőség ez esetben rendkívül fontos.

Tavaly ősszel kísérleti jelleggel a Magyar Posta is sugárzott az URH-adón számítógépes programokat — a radiosoft nevű új szolgáltatás előkészítéseként. A megnyíló távlatok igen szélesek — mindenkinek szárnyalhat a fantáziája! —, de kézenfekvő például, hogy a lapunkban megjelenő programokat meghatározott időpontban rádióadással is lehetne terjeszteni, így mentesülnének olvasóink a begépelés fáradságos, sok hibával és azok keresésével járó munkájától.

Pascal-bővítések

Világszerte növekszik a Pascal nyelv népszerűsége. Az IBM PC-vel kompatibilis gépeken futó Turbo Pascal széles körű alkalmazását követi az itthoni tendencia: ebben a gépkategóriában hazánkban is az alkalmazói szoftverek jelentős része készül ma már a Turbo Pascal fordítóval.

E programnyelvet előnyös tulajdonságai kedveltették meg a felhasználókkal. A strukturált programozás közvetett támogatásával könnyebb a programokban való hibakeresés, a későbbi módosítás; a konkrét végrehajtást a nyelv saját szerkesztője segíti. A programfordítás rendkívül gyors, maga a programírás is szaporább: sok programozó esküszik például arra, hogy egy 1000 soros COBOL programot Pascalban 200-300 sorban megír.

A Turbo Pascalban programozóknak több hatékony szoftvereszköz, például az Open Acces integrált adatbázis-kezelő és a Turbo Acces indexfájlkezelő rendszer is rendelkezésére áll. Az ilyen komplex szoftvereken kívül ma már szerte a világon jelennek meg a Turbo Pascalhoz forrásszinten beépíthető, adott felhasználói rendszerek fejlesztésénél jól hasznosítható eszközök is (például Nyugaton forgalmaznak egy grafikai célú gyűjteményt). Hazánkban a Számszöv Kisszövetkezet kezdeményezett ilyen irányú fejlesztéseket, főképpen a többfelhasználós rendszereket kiszolgáló programok készítésének megkönnyítésére. Alaptermékük a T RANDOM, mely rekordszintű adatvédelemmel javítja a többfelhasználós rendszerek kiszolgálását, s többek között képernyőkezelővel és fájlnyilvántartóval is ki lehet egészíteni. Ezen kívül számos autonóm működésű, a Pascal programozást segítő eszközt kínálnak, például nyomkövető-tesztelőt, programméret-optimalizálót, maszkgenerálót és lekérdezőt, sőt magyar ékezetes Turbo Pascalt is.

Világító billentyűzet

Már kezdenek megjelenni a hazai kirakatüvegeken is az érintéses billentyűzetkivezetések. Ezeknek egy új nemzedékét jelenti a Szint Kisszövetkezet szabadalma: olyan világító billentyűzet, melyet közönséges zsírkrétával lehet feliratozni, s a ráírt szöveg, ábra a zsírkréta árnyalatával megegyező színben fénylik. Még jobban megragadja a járókelők figyelmét, ha a zsírkrétát egy ügyes grafikus kezébe adják!

A gyakran változó, gyors feliratozásra is kitűnően alkalmas ez a módszer: ez azt jelenti, hogy több szoftverterméket is futtathatunk a kirakatban, naponta akár többször is cserélve a kirakatüvegre kivezetett billentyűk értelmezését.

Az eljárás természetesen nem csupán az aktív billentyűzet, hanem passzív hirdetőtáblák kivitelezésére is alkalmas. Különösen a diszkrét világítású helyeken érvényesül a tündöklő tábla; például az elegáns éttermekben hangulatosan hívhatja magára a figyelmet a konyhafőnök ajánlata.

Földrengésjelzés

Befejezték az ország rádiótávközlésű, automatikus földrengésjelző rendszerének kialakítását. A 17 millió forintos beruházás első szakaszában a piszkéstetői és a budapesti, most pedig az újkígyósi, valamint az ózdi állomást építették meg.

Az állomások sorra kapcsolódtak be az országos hálózatba, de eddig csak analóg jelek küldésével szolgáltattak adatokat. A most beszerelt mikroprocesszoros elektronika által már nem csupán küldenek adatokat, hanem azok fogadására és tárolására is képesek. A korszerűsítés eredményeképpen az év elejétől kezdve ezek az állomások már emberi felügyelet nélkül működnek. Automatikusan azonnal jelzik és közvetítik a budapesti obszervatóriumnak az ország és a világ bármelyik részén keletkező földrengéseket. Itt a szeizmogramokat számítógéppel folyamatosan értékelik, így gyorsan megállapíthatják (becsülhetik) a rengések időpontját, helyét és erősségét.

Forgalomirányítás

1986 őszén kezdte meg kísérleti üzemét az a rendszer, amelyet a Közlekedéstudományi Intézet kutatói dolgoztak ki Székesfehérvár jelzőlámpás csomópontjainak központi ellenőrzésére és irányítására.

A rendszerkidolgozás első szakaszában a jelzőlámpák központi felügyeletét, működésük folyamatos figyelemmel kísérését oldják meg számítógép segítségével: így szükségtelen a jelzőlámpás csomópontok naponkénti bejárása, megvizsgálása.

A munka második szakaszában, az idén kapcsolódik be a

számítógép a lámpák tényleges irányításába. Érzékelődetektorokat építenek az útburkolatba, melyek folyamatosan jelzik a számítógépnek a forgalmat. A gép az adott csomópontra kidolgozott programok közül automatikusan választja ki a pillanatnyi igényeknek megfelelő, optimális változatot, s ennek alapján vezérli a lámpákat. Ezáltal a közlekedés még a csúcsforgalom idején is zavartalan lehet.

A székesfehérvári számítógépes forgalomirányító rendszer paraméterezhetően készül, így alkalmas lesz más, hasonló nagyságú városok forgalmának irányítására is.

AT-kompatibilis

Megjelent a hazánkon kívüli szocialista országok első, IBM PC AT-vel kompatibilis gépe, a bolgár Intelat. A szófiai Információs Rendszerek és Rendszermérnöki Szolgáltatások nevű termelési egyesülés által készített gép hétféle tipuskonfigurációban kerül forgalomba. A legkisebb tipikus AT-kiépítés (640 k operatív tár, 20 Mbájt Winchester-tár), a legnagyobbat pedig a grafikus alkalmazásokhoz szükséges speciális perifériákkal is ellátták.

Elektronika Indiában (Eddig még kimaradtunk . . .)

India elektronikai ipara az ország nemzeti jövedelmének mintegy 0,8 százalékát képviseli; a fejlett országokban ez az arány 1—10% között mozog. India kormánya az ország 1985—90-ig tartó ötéves tervének végére ezt 3 százalékra kívánja növelni.

A fejlesztés során a hangsúlyt eleinte a licencvásárlásokra és a külföldi gyártástechnológiák átvételére helyezik. Ennek során szorosan együttműködnek a szocialista országokkal is. Az indiai elektronikai minisztérium és a kereskedelmi minisztérium különböző elektronikai vonatkozású, kétoldalú egyezményeket kötött Csehszlovákiával, az NDK-val, Romániával és a Szovjetunióval.

1 kontra 64 Mbit

A Siemens és a Philips már több mint egymilliárd NSZK márka állami támogatást kapott a nagy kapacitású tár kifejlesztéséhez. Az első 1 Mbit kapacitású tár előállításához részben felhasználják a Toshiba technológiáját, és még az idén piacra viszik ezt a terméket. Ez a határidő azonban nemzetközi viszonylatban túlságosan késői. Már most tapasztalni lehet ugyanis, hogy a 256 kbit kapacitású félvezető tárak élettartama rövidebb lesz, mint ahogy eredetileg számították, sőt a még nem mindenütt beszerezhető 1 Mbit kapacitású RAM tárak iránti érdeklődés ideje is várhatóan rendkívül rövid lesz: minden valószínűség szerint Japán már 1989-től kezdve forgalomba fogja hozni a 64 Mbit kapacitású tárakat.

Kerüld ki a Piccadillyt!

A brit közlekedési minisztérium laboratóriumában kisszámítógépet fejlesztettek ki, amely személyautókba, teherautókba szerelve alkalmas a forgalmi dugók jelzésére, s útmutatást is ad a vezetőnek arra, hogy miként kerülje ki a zsúfolt közutakat. A rendszer lényege, hogy az autóutak mentén felszerelt különleges érzékelők figyelik a forgalom alakulását, és mérésük eredményéről rádiójelekkel tájékoztatják az autókban lévő számítógépeket, amelyek a kapott információk alapján azonnal jelzik a legkevésbé zsúfolt útvonalat. Az idén már kapható készülék kb. 150 fontba kerül, viszont — mint a brit közlekedési minisztérium illetékesei kifejtették — tömeges alkalmazásával legalább 100 millió fontot lehet nyerni üzemanyagban, a balesetek elkerülésével és mindenekelőtt a megtakarított idővel.

Virágzó szoftverpiac

Az USA-ban továbbra is kifejezetten kedvező a számítástechnikai szoftver piaca. 1985-ben a 30%-os növekedéssel 12,5 milliárd USD értékű forgalmat értek el. Az 1986. évi előzetes adatok szerint 16,4 milliárd dollár értékű szoftver kelt el, ami az előző évhez képest 31%-os többletnek felel meg. Különösen a személyi számítógépi szoftver területén ível fölfelé a fejlődés: üteme 55%-os, és értékben eléri a 6,2 milliárd USD-t. A szoftverforgalom negyedét az IBM bonyolítja le.

Univerzális illesztő (magyar!)

Az ELTE Általános Technikai Tanszéke — az OMFB támogatásával és elsősorban oktatási célokra — kifejlesztette a technoMIR nevű, számítógéptípustól független, univerzális illesztőrendszert. Feladata a számítógép és a külvilág — a fizikai, kémiai, biológiai és technikai környezetünk — közötti kapcsolatteremtés. A technoMIR rendszer egységei valamilyen sajátos funkciót látnak el: a külvilágból érkező információt alakítják a számítógép által feldolgozható adatformára, vagy éppen ellenkezőleg, a gépből adat formájában érkező parancsot formálják a fizikai-kémiai folyamat befolyásolására alkalmas alakra. Az egységek külsőre azonos kialakításúak, és könnyedén egymásba dugaszolhatók. Az összecsatlakoztatott modulok fajtáját az adott kísérlet vagy a vezérelni, szabályozni kívánt folyamat határozza meg. Sorrendjük, számuk tetszőleges, így akár egyszerre több folyamatba is beavatkozhat ugyanaz a számítógép.

A technoMIR kiválóan alkalmazható laboratóriumokban, sőt ipari alkalmazása is vonzó lehet, mivel a mintegy 4000 forintos átlagárú modulok az ipar számára árusított illesztőkhöz képest rendkívül olcsók. A rendszer a Z80 mikroprocesszorral működő mikroszámítógépekkel együtt közvetlen alkalmazható, más típusokhoz pedig a gyártó Electroscan Kisszövetkezet illesztőmodult szállít.

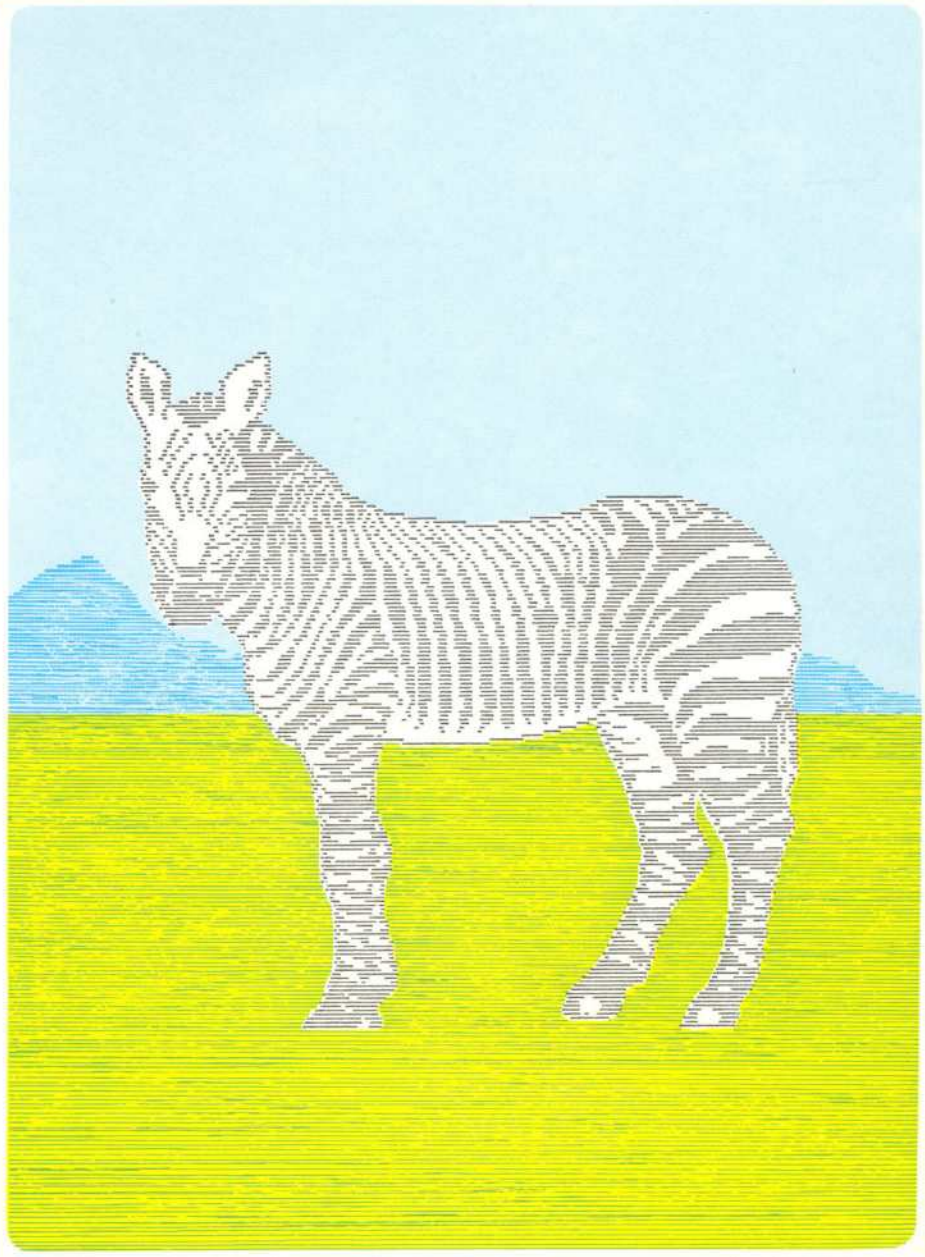
Az elmúlt tanévben az Országos Pedagógiai Intézet 20 db, egyenként 10 modulból álló technoMIR készletet osztott ki az iskoláknak tapasztalatgyűjtési, módszertan-készítési kötelezettséggel. Ebben a tanévben a kedvező vélemények alapján különböző szintű oktatási intézmények — az általános iskoláktól az egyetemekig — csupán decemberig már 200 készletet vásároltak. A technoMIR sikeres már a debreceni Kossuth Lajos Tudományegyetemen és több tanárképző főiskolán is, például Szegeden, Szombathelyen és Pécsett.

Programvédelem PC DOS és MS—DOS alatt

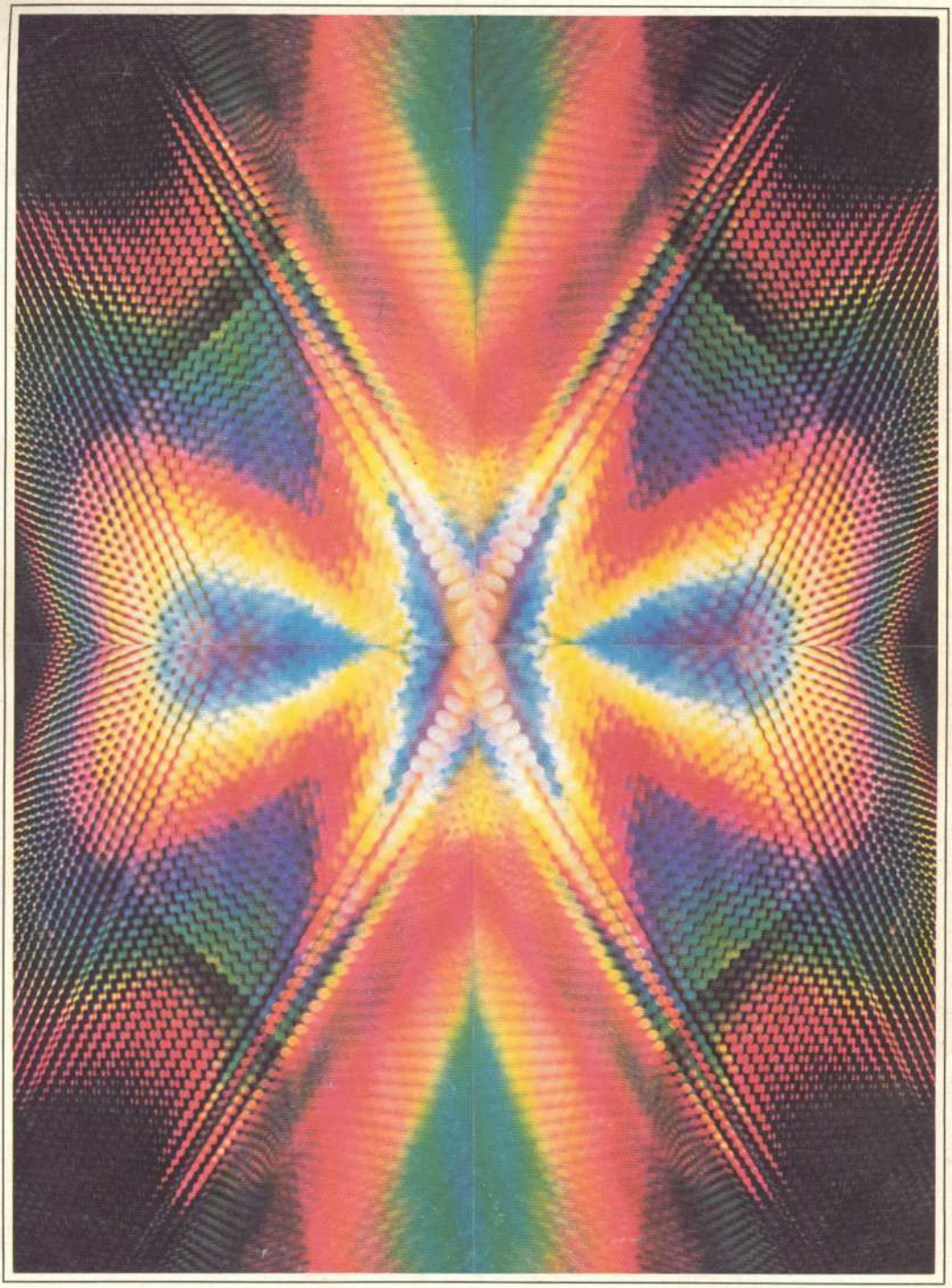
Hazánkban még nem bővelkedünk a szoftvervédelemre szolgáló ajánlatokban. Ezért is jelent újdonságot a Softinvest által forgalmazott EltGUARD nevű programvédelmi rendszer. Ez valamennyi PC DOS, illetve MS—DOS operációs rendszerrel működő IBM vagy vele kompatibilis személyi számítógépen futtatható program (.EXE és .COM fájlok) utólagos másolás, illetve illetéktelen felhasználás elleni védelmére alkalmas.

Az EltGUARD-dal védett programok, illetve a programok futtatásához szükséges kulcslemezek semelyik ismert másolóprogrammal (COPYIPC, COPYWRITE, SUPERCOPY stb.) sem másolhatók eredményesen, és a kulcslemezek hiányában az eddig bevált védelemelhárító rendszerekkel (ZERODISK, NOKEY, UNGUARD, NOGUARD stb.) sem indíthatók el.

Az EltGUARD védelmi rendszer alkalmas egy vagy több program azonos vagy egymástól elkülönülő védelmére is. Jól használhatják mindazok, akik az általuk készített vagy forgalmazott szoftver valamennyi példányát óvni akarják a másolástól.



A ZEBRÁT AZ RAJZOLÓGÉPE

KÉSZÍTETTE:

Cikkünk a 16. oldalon